A RAND NOTE

Guide for the Management of Expert Systems Development: Additional Appendixes

Iris Kameny, Umar Khan,
Jody Paul, David Taylor

August 1989

RAND

The research described in this report was sponsored by the Assistant Secretary of Defense (Production and Logistics) under RAND's National Defense Research Institute, a federally funded research and development center supported by the Office of the Secretary of Defense, Contract No. MDA903-85-C-0030.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER N-2970-P&L	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Guide for the Management of Expert Systems Development: Additional Appendixes		5. TYPE OF REPORT & PERIOD COVERED interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) I. M. Kameny, U. Khan, J. Paul. D. G. Taylor		8. CONTRACT OR GRANT NUMBER(s) MDA903-90-C-0004
9. PERFORMING ORGANIZATION NAME AND ADDRESS RAND 1700 Main Street Santa Monica, CA 90401		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Assistant Secretary of Defense for Production & Logistics Dept. of Defense, Washingotn, DC 20301		12. REPORT DATE August 1989
		13. NUMBER OF PAGES 85
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No Restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Information Systems
Computer Programs

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

See reverse side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This Note, a companion publication to R-3766, contains two appendixes that complement that report. The first provides assistance in selecting the most appropriate tool for a particular expert system development project; the second describes logistics expert system applications listed in alphabetical order by application name.

A RAND NOTE

N-2970-P&L

Guide for the Management of Expert Systems Development: Additional Appendixes

Iris Kameny, Umar Khan,
Jody Paul, David Taylor

August 1989

Prepared for the
Assistant Secretary of Defense
(Production and Logistics)

RAND

PREFACE

This Note is a companion publication to RAND report R-3766-P&L, *Guide for the Management of Expert Systems Development*, by Iris Kameny, Umar Khan, Jody Paul, and David Taylor, July 1989. The Note contains two appendixes that complement that report. This research was sponsored by the Assistant Secretary of Defense (Production and Logistics). It was conducted under the Information Processing Systems Program within The RAND Corporation's National Defense Research Institute, a federally funded research and development center sponsored by the Office of the Secretary of Defense.

The Office of the Secretary of Defense is preparing a concise *Quick Reference Guide* to accompany this important reference. This companion Guide will emphasize the differences between traditional software development and expert systems development for program managers and other developers. Using detailed references to the RAND Guide, the *Quick Reference Guide* will make it easy to find, understand, and use the wealth of knowledge incorporated in the RAND document. This Guide will be available early in 1990 from the Office of the Assistant Secretary of Defense (Production and Logistics), Directorate for Automation Support and Technology.

CONTENTS

FIGURES

TABLE

Appendix C

TOOL EVALUATION AND SELECTION

This appendix provides assistance in selecting the most appropriate tool for a particular expert system development project. A strong foundation in the usage of any specialized tool (e.g., chemical assay equipment or an expert system shell) is prerequisite to tool selection. Familiarity and reasonable facility with expert systems technology and knowledge engineering are necessary prior to applying the techniques presented in this appendix.

This appendix outlines a selection method that was adapted from one developed at The RAND Corporation for expert system tool evaluation [Rothenberg 1987]. It is applicable after the decision has been made that such a tool is needed and the characteristics of that need are known. In particular, this involves (1) understanding the problem that shows potential for applied expert system technology and (2) choosing the best solution alternative.

The term *expert system* means a system built using a knowledge-based approach to software development that applies expert knowledge to solve difficult real-world problems. The tools for building such systems should be called *knowledge-based system building tools*, although they are commonly known as *expert system building tools, expert system tools,* or *expert system shells*. For simplicity, in this appendix we use the term *tool* to mean any piece of software intended to help design, build, deliver, or maintain an expert system. A tool comprises not only a specific software environment but all aspects of the software entity and its use, including: training, documentation, ease of use, vendor support, and cost.

Note that the tool selection methodology presented in this appendix concerns building and using expert systems and does not consider standard computing solutions. Specifically, we do not address the use of specialized programming environments, such as expert system shells, for building conventional types of applications. Expert system shells belong to the growing class of 4th Generation Languages (4GLs), along with spreadsheet programs, database front-ends, and the like. Just as every item built using an aircraft fastener is not necessarily an aircraft, nor everything built using a furniture saw a piece of furniture, not every piece of software built using an expert system shell is an expert system. Shells are attractive because they provide the ability to rapidly generate applications using a well-designed developer interface and a simplified higher-level of programming that is closer to specification writing than detailed procedural coding. The resulting applications can take advantage of extensive end-user interface supports, such as menus, windows, graphic displays, etc.

The use of shells as programming environments is an exciting area of software development and the topic of much discussion in the software engineering community [Boehm 1988, Tracz 1988, Brooks 1987, Kolodziej 1988]. Our focus is the development and deployment of expert systems and as such does not deal with the use of 4GLs as productivity enhancement tools for generating conventional solutions. The tool selection process for non-expert-system applications is not novel. The concerns of software designers and engineers in choosing an implementation language and environment apply to these new languages and development environments just as they do to other languages and environments.

C.1 APPLYING THE RAND TOOL EVALUATION METHODOLOGY

The method developed at The RAND Corporation for tool *evaluation* lends itself readily to use for tool *selection.* This is due in large part to recognition during the tool evaluation study that an absolute and context-free evaluation of tools is not meaningful. Rather, their strengths and weaknesses must be viewed in the particular context in which they will be applied. We describe a framework of criteria and a method for selecting an expert system tool for a particular problem. The framework is really only a starting point that must evolve as the field and the tools mature. The method is general and designed to apply to any problem and set of potential tools in the foreseeable future. The method applies the framework to select the most appropriate tool (or tools) for a given task. The process of tool selection can never be entirely mechanical—the skills and knowledge of the expert system developer will significantly affect the outcome. Tool selection is really a matching process: matching an expert system tool with the intended use. Three key match issues concern the problem, the intended system, and the development team.

Problem:	Does the tool have the features suggested by the needs of the problem?
System:	Does the tool have the features suggested by the needs of the system?
Developer:	Does the tool provide the developer with the necessary power and sophistication?

We next describe the tool evaluation framework and the method for applying that framework for tool selection. Keeping these three questions in mind should help focus on the key aspects of the following discussion.

C.2 TOOL EVALUATION FRAMEWORK

The framework presented here is designed to apply to a broad variety of tasks and to both existing tools and new tools; therefore it includes many criteria. These criteria can be greatly pruned by identifying items of particular significance to a specific project. Furthermore, the dimensions can be prioritized. This pruning and prioritizing make tool selection manageable for any given project. The specific needs of each particular application must be considered uniquely—e.g., the concerns for the very large, multiple expert system B1-B CITS are not the same as those for the well-defined, stand-alone Airlift Allocation expert system. (See Appendix D for descriptions of these and other applications.) Tool evaluation involves five distinct dimensions, illustrated in Figure C.1. These dimensions are used in the tool selection process as follows:

> Given the relevant application characteristics, apply metrics by means of assessment techniques to evaluate particular capabilities of a tool in particular contexts.

We briefly describe each of these dimensions below. More complete coverage may be found in Rothenberg [1987]. The method itself is described in Section C.3.

C.2.1 Application Characteristics

Application characteristics are comprised of *problem characteristics*, *usage* (or *target environment*) *characteristics*, and *project characteristics*. These represent the impact of the application on tool selection. Certain types of solutions are suggested by the application characteristics, which further suggest what features are needed in the expert system building tool.

C.2.1.1 Problem Characteristics

Figure C.2 illustrates the types of problem characteristics that affect the choice of a tool. The kinds of knowledge and processing that characterize a problem domain may provide useful criteria for choosing among tools. For example, an inventory control expert system requires the expression of domain concepts such as rate of consumption and the use

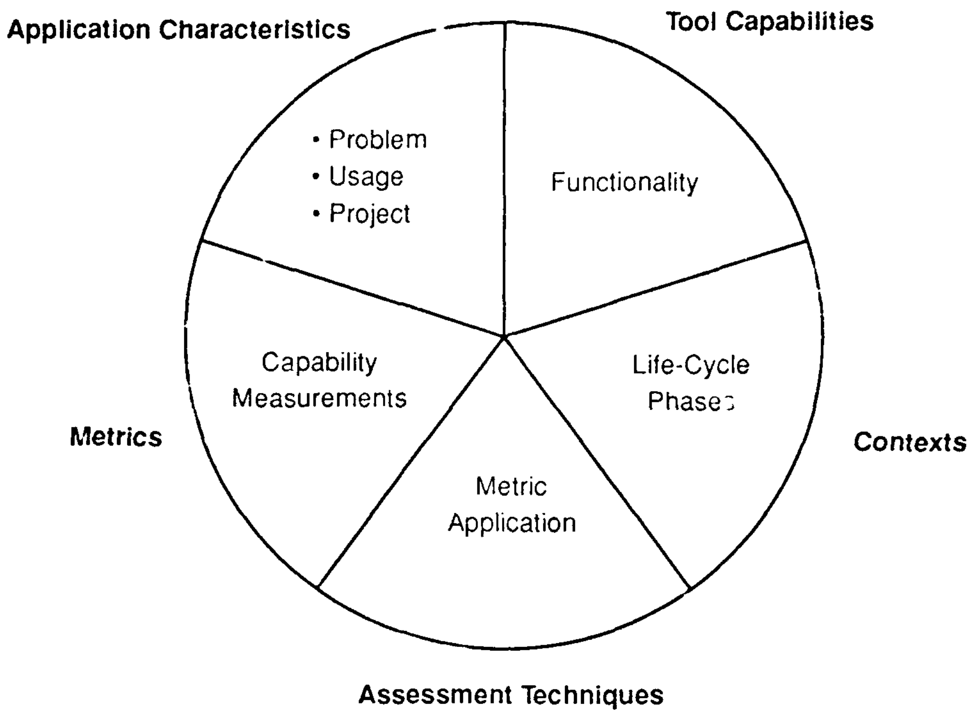

Figure C.1—Tool evaluation framework

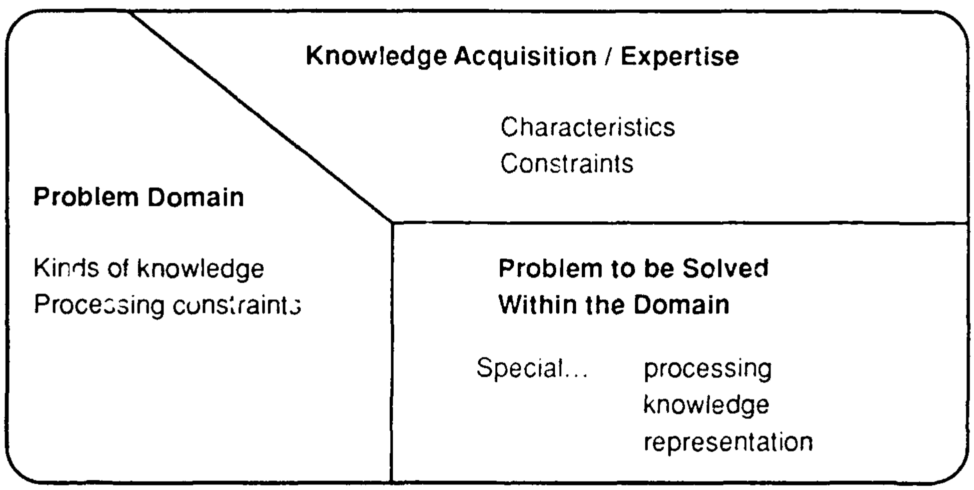

Figure C.2—Problem characteristics

of computational techniques such as forecasting. In some cases, a tool may incorporate specific mechanisms and knowledge oriented toward a particular domain, such as a language that recognizes domain-specific jargon. Often, the domain suggests requirements for specific tool features. For example, financial or legal applications may require strict accountability; applications that involve simulation may require spatial reasoning; and process control applications may have real-time and critical reliability requirements. The particular problem to be solved within the domain may involve special kinds of knowledge, processing, or representational requirements that may lead to special requirements for a tool. The problem may also establish requirements and constraints for the capacity, performance, and delivered cost of the target system, as well as for its availability, reliability, robustness, and maintainability. Many problems require that a target expert system communicate and integrate with existing databases, software, and hardware. These integration issues are often crucial because of the difficulty of building such interfaces. Additional problem attributes that should be considered include expected complexity and storage requirements; operational constraints such as execution speed, real-time requirements, compatibility requirements, physical environment, hardware portability requirements, or the need for some proof of correctness; and formal properties, such as decomposability, the degree to which the problem lends itself to algorithmic vs. heuristic solution, and symbolic or numeric requirements. A set of important considerations comprises special characteristics or constraints that apply to the knowledge acquisition or sources of expertise in the application domain. Such concerns include the need for multiple knowledge sources and the coordination of multiple knowledge bases. Other concerns of this nature are related to the development environment and team and are discussed below in the section on "Project Characteristics."

C.2.1.2 Usage Characteristics

Characteristics of the intended usage of an expert system—its target environment—are illustrated in Figure C.3. The target environment for an expert system determines its delivery hardware and its need to integrate with existing hardware, software, databases, and networks. It also establishes requirements and constraints for the capacity, performance, and delivered cost of the target system, as well as its availability, reliability, robustness, and maintainability. Similarly, the characteristics of the expected end-users of the target expert system (their level of experience with computers, their domain expertise, their educational background, etc.) determine user interface and explanation requirements for the target system and therefore for the tool.

C.2.1.3 Project Characteristics

Project characteristics include characteristics of the expert system project, its development environment, and its development team (shown in Figure C.4). General development constraints, such as time, money, personnel, and hardware, strongly influence the choice of a *type* of tool—e.g., a programming language, a basic knowledge engineering language, or a fully integrated shell.

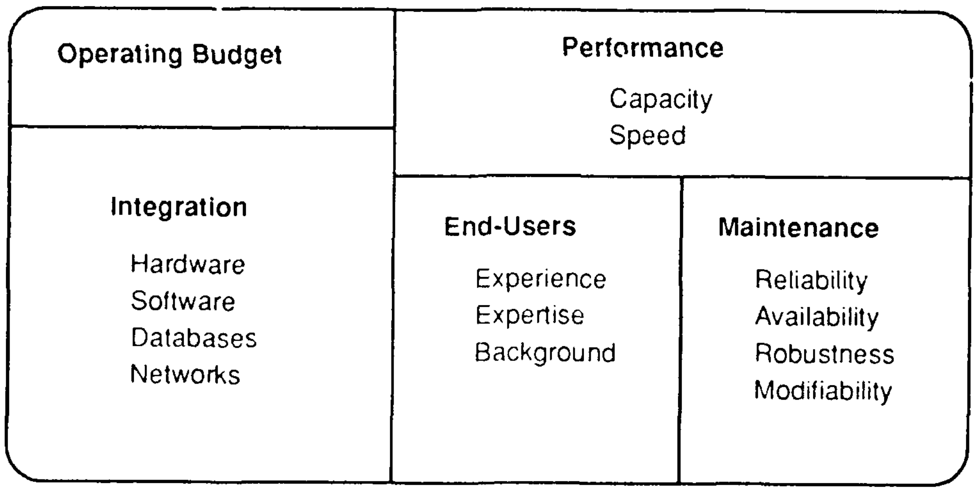

Figure C.3—Usage (target environment) characteristics

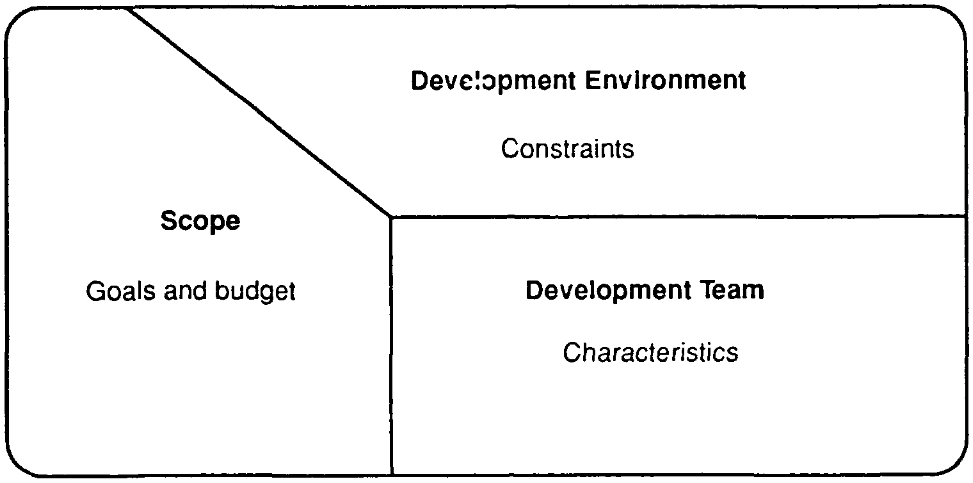

Figure C.4—Project characteristics

The scope, goals, and budget of an expert system effort are among the most important factors in determining what kind of system will be built, and therefore what is required of the tool. Closely tied to scope is the question of budget. Cost is often an overriding (or vetoing) factor in deciding on a tool. The *development environment* delimits the hardware and software on which a tool must run as well as the network and database interfaces it must provide during development. These factors may be given *a priori* (the items already in place) or a project may have some control over them (e.g., the freedom to choose hardware based on the preferred tool). The composition of the *development team* (whether preexisting or contemplated) must also be taken into account. The size of the team and the members' background, preferences, and previous experience are important. The characteristics of the knowledge engineering part of the development team will affect the kinds of support required for knowledge acquisition.

C.2.2 Contexts

Each context in which a tool can be used (illustrated in Figure C.5) is named for the development life-cycle phase in which it is dominant, although a given context may apply across several development stages. For example, tool requirements for "deployment" may apply early in the conceptual design stage of a project. Delimiting contexts as a fixed set of points must not obscure the transitions between these points. The transitions between development phases can be just as important as the phases themselves. The relevant contexts for a given development effort are derived from the application characteristics, primarily the project scope and the problem to be solved.

Conceptualization emphasizes a tool's support for conceptualizing, formalizing, and decomposing a problem, and for identifying and organizing key concepts. *Definition / Design* emphasizes a tool's facilities for guiding rapid development and quickly trying different approaches, representations, and alternative implementations. *Development* considers a tool as it is used to develop an expert system, emphasizing the support for software development (including debugging facilities, configuration management, etc.). *Deployment* concerns a tool's facilities for porting from the development environment to the delivery environment and for integration and end-user interface support. *Post Deployment* emphasizes a tool's support for the performance, maintainability, and supportability of the target expert system in its delivery environment.

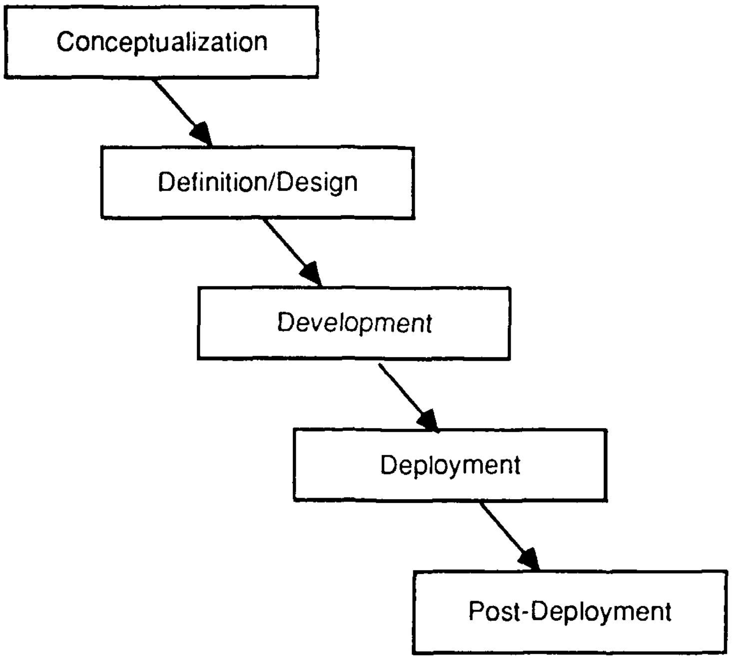

Figure C.5—Tool usage contexts

C.2.3 Tool Capabilities

Tool capabilities reflect the functionality of expert system tools. The tools currently on the market provide many features supporting a wide range of capabilities. We focus on the capabilities of a tool rather than the specific features it provides for achieving those capabilities. This emphasizes the functionality of the tool rather than the specific implementation of that functionality. This dimension is the most dynamic. As expert system technology and its tools continue to evolve, the list of relevant capabilities also evolves and expands. Examples of current capabilities and representative features that support them are shown in Table C.1. This list is not exhaustive, and the examples of supporting features are merely illustrative.

C.2.4 Metrics

Metrics are measures of tool capabilities. They are applied to particular capabilities of a tool using the assessment techniques described below. The following aggregated metrics capture most of the relevant qualities of a tool:

- Cost
- Flexibility
- Extensibility
- Clarity
- Efficiency
- Vendor support

Cost includes hidden expenses such as training and integration, as well as the purchase price and support costs of a tool. Resources consumed (costs) may be money, person-power, machinery, supplies, computation used, elapsed time, etc. The effects of cost are felt throughout the life cycle of a system, but its relevance peaks at the transitions between development life-cycle phases, where a project either commits to switching tools or stays with

Table C.1

TOOL CAPABILITIES AND SUPPORTING FEATURES

Capability	Examples of Supporting Features
Arithmetic processing	Arithmetic operators, extended floating point
Certainty handling	Certainty factors, fuzzy logic
Concurrency	Distributed processing, parallel processing
Consistency checking	Knowledge base syntax checking
Documenting development	Assumption/rationale history, code/data annotation
Explanation	Execution trace, knowledge base browsing
Inference and control	Iteration, forward/backward chaining, inheritance
Integration	Calling other languages, interprocess calls
Internal access	Tool parameter setting functions, source code
Knowledge acquisition	Rule induction, model building aids
Knowledge base editing	Structure editors, graphic rule lattice
Life cycle	Tool support for target system life cycle support
Meta-knowledge	Rules controlling inference, self-organizing data
Optimization	Intelligent look-ahead, caching, rule compilation
Presentation (I/O)	Text, graphics, windows, forms, mouse
Representation	Rules, frames procedures, objects, simulation

what it has. *Flexibility* includes representational power (data structures and reasoning mechanisms), adequacy to the given task or tasks, breadth of applicability, and sophistication. *Extensibility* includes breadth of applicability, access to system parameters, the ease with which system parameters or functions can be overridden, ease of integration, portability, and scalability. *Clarity* includes the ease of understanding and using a tool, cognitive efficiency (i.e., how many concepts must be kept in mind to use the tool), maintainability, modularity, learnability, coherence of the tool's features, and how appropriately the tool responds. *Efficiency* includes speed of response and utilization of computational and memory resources. During development, the efficiency of a tool manifests itself in terms of compilation speed, response time, and knowledge base memory requirements, all of which affect the development process. However, the ability of a tool to produce an efficient target system tends to overshadow the efficiency of the tool itself, although there are cases in which it may be crucial to the start of a project. *Vendor support* includes vendor philosophy, training, system availability, reliability, portability, and robustness.

C.2.5 Assessment Techniques

Assessment techniques are the means bv which metrics are applied. For some metrics, assessment techniques seem obvious and straightforward. Evaluating the initial cost of a tool, for example, may be simply a matter of asking for a quote or it may be investigating whether a copy has already been purchased for the organization and a site license is available. However, even in this highly quantitative realm, evaluating learning cost, long-term cost, or cost-effectiveness is often far from trivial. Similarly, feature comparison is often performed by simply asking whether a tool has a given feature. Unfortunately, this kind of comparison can be quite misleading, because it is limited by the depth of meaning behind the labels assigned to features. There are currently few objective ways of applying metrics to a tool prior to using it. Any measurement technique that claims to be objective is subject to suspicion, especially if it is also quantitative. For example, performance benchmarks are considered to be unreliable by tool developers and users alike because they are too easily distorted by implementational shortcuts or by comparing incommensurable items (e.g., showing relative speeds for processing "rules," where the granularity and power of a rule may vary widely among tools).

In choosing a tool we must often rely on anecdotal evidence from colleagues who have applicable experience, as represented in Appendix D of this document. Such experience and personal advice ranks high among users as a valuable assessment technique. Our approach to the application of metrics is to suggest a number of assessment techniques that do not in general produce quantitative measures, but instead produce textual results. When a metric is applied using one of these techniques, the result will require intelligent interpretation. However, this is an advantage, since it gives the decisionmaker valuable information. The assessment techniques that seem the most promising are:

- Direct comparisons,
- Benchmarks,
- Interviews, questionnaires, and personal advice, and
- Libraries of case studies and development efforts.

Direct comparisons among tools can be valuable if they focus on corresponding capabilities of tools and explicate the ways in which these capabilities differ. Comparison with an abstract standard or with "conventional" AI approaches may also be useful. *Benchmarks* are well-formulated statements of more-or-less fragmentary problems. The term does not mean a quantitative performance measure or a feature-based comparison (such as how many rules a tool can process per second). Benchmarks can be small (e.g., testing how a tool can represent a class hierarchy) or large (such as the classic "spills" problem).[1]

Small benchmark problems can be used to compare specific capabilities of tools, provided they are interpreted on the basis of the style of their solutions rather than their performance. Implementation of larger benchmarks using a number of different tools may be warranted prior to a major tool commitment. To use benchmarks, it is necessary to formulate problems that are specific but do not require a particular implementation. Some illustrative benchmarks are presented in Rothenberg [1987]. Interviews, questionnaires, and personal advice from other developers, tool-users, and colleagues can provide a wealth of useful information. A small library of case studies and expert system development efforts is included in Appendix D.

C.3 METHODOLOGY

An overview of the RAND tool evaluation methodology, modified for tool selection, is shown in Figure C.6. The essence of this evaluation technique may be summarized as:

> Given the relevant application characteristics, apply metrics by means of assessment techniques to evaluate particular capabilities of a tool in particular contexts.

In the following sections we discuss each of the steps in the evaluation process. Since the framework dimensions have been discussed in the previous section, the way in which that information is used is described.

[1] Described in Kolodziej [1988], the "spills" problem is a case study in knowledge engineering: an expert system is needed to help consult with regular workers or to augment the limited experience of off-shift workers facing the difficult task of responding to an accidental spill of an oil or hazardous chemical at Oak Ridge National Laboratory.

C.3.1 Determine Application Characteristics

As the first step, the system designer must seek to answer the question, "What is the nature of the expert system to be built?" by determining the characteristics of the intended application. Application characteristics are comprised of problem characteristics, usage characteristics, and project characteristics (illustrated previously in Figures C.2, C.3, and C.4). Certain types of solutions are suggested by these application characteristics, which further suggest what features are needed in the expert system building tool. The analysis can be broken down into responses to three more focused questions reflecting the problem environment, the sustaining environment, and the system development process itself (which spans both environments):

- "What is the nature of the problem to be solved by the expert system?"
- "How will the expert system be used?"
- "What resources are available for developing the expert system?"

The main distinction between this analysis and that described earlier in this appendix is that the characteristics are to be expressed in terms of the chosen technology—expert systems. Application characteristics should be weighted as to certitude and importance. The scope and goals of a project will determine which characteristics represent obligatory requirements and which ones are negotiable.

C.3.2 Identify Relevant Contexts

The contexts targeted for a project supply the other major factor for determining required tool capabilities. It is particularly important here to be realistic about which phases of development are to be undertaken with the tool(s) to be chosen. Targeting only the initial exploratory or prototyping phases may lead to choosing a tool that encounters a dead end if the development process is extended further. On the other hand, targeting all phases when only some of them are likely to be undertaken will over-constrain the selection process.

C.3.3 Identify Discriminating Metrics and Assessment Techniques

Certain metrics, such as cost, may have high discriminating (or vetoing) power in choosing a tool. If such metrics exist for a project, they should be identified, along with the best available assessment techniques for applying them at this stage. This early discrimination helps improve the manageability of the tool selection process.

C.3.4 Prune and Prioritize Each Framework Dimension

Each dimension of the framework should be pruned to eliminate irrelevant or inapplicable criteria. The remaining items should be prioritized or weighted. At this point, after the full investigation of the problem and environment, enough should be known about the application characteristics and contexts to prioritize each framework dimension separately. Further filtering may occur throughout the remaining stages of tool selection, for example, after tool capabilities are more fully assessed or the set of candidate tools has been established. The filtration process is also dynamic. For example, if cost filtering has already resulted in a set of candidate tools whose costs are very similar, cost becomes an ineffective metric for following stages. In most cases, strict prioritization will be inappropriate. For example, cost might be a more important metric than efficiency, but a *very* efficient tool might well be worth *some* extra cost. Thus, "prioritization" should be thought of in the general sense of assigning weights to items in a dimension to reflect importance relative to the other items in that dimension. (Note that assessment techniques are prioritized somewhat differently, dependent on their availability, applicability, believability, timeliness, and cost.)

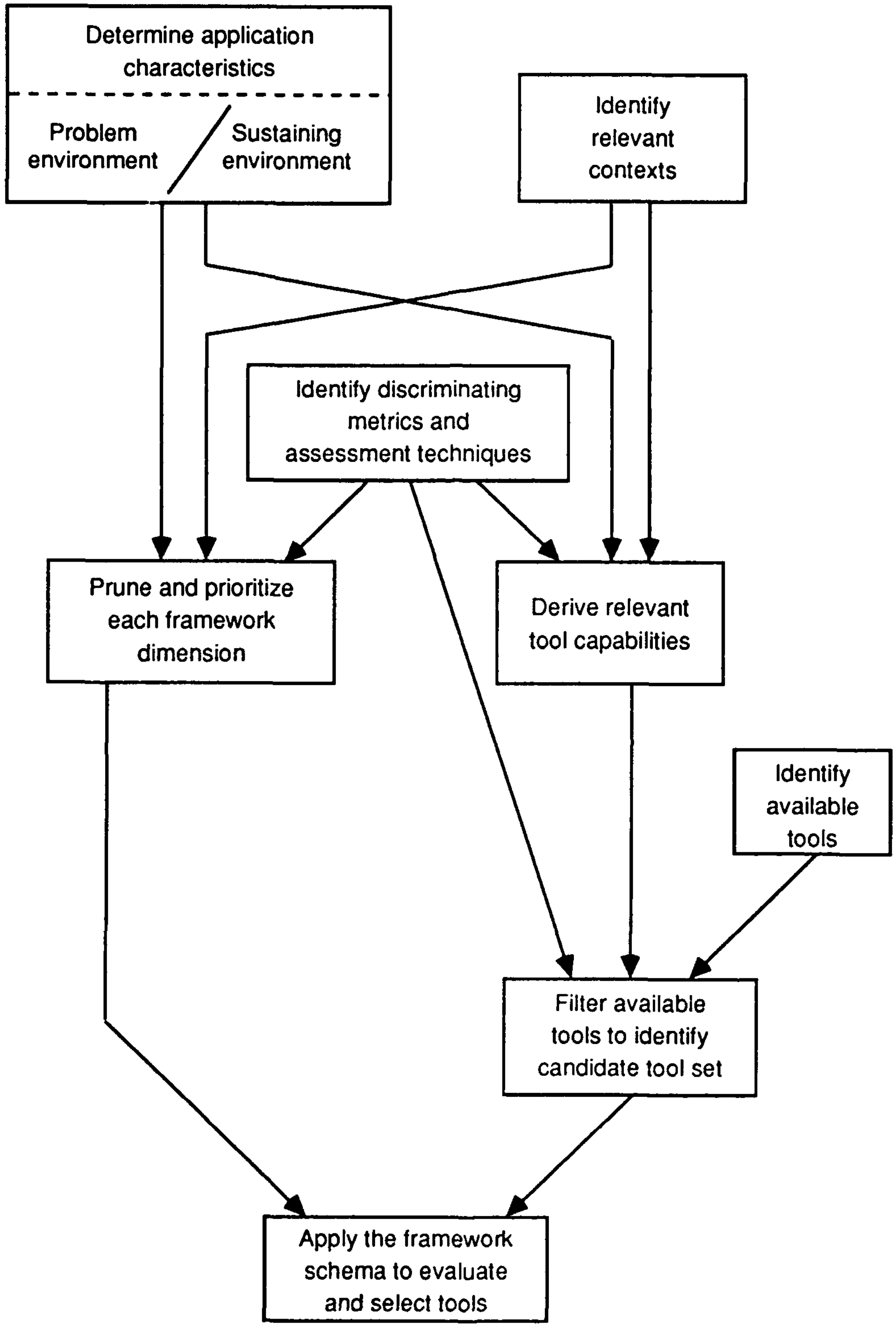

Figure C.6—RAND tool evaluation methodology

C.3.5 Derive Relevant Tool Capabilities

The tool capabilities are derived from the application characteristics and the relevant contexts. It is important at this stage to weight these capabilities along a square from *required* to *desired*, for use in filtering the available tools in a later step. These weights are derived from the application characteristic weights determined in the first step.

C.3.6 Identify Available Tools

This step also typically involves some implicit filtering, because it is difficult to find all available tools. It is desirable to make as complete a survey as possible, filtering explicitly in the following step instead. A survey of expert system tools is a good place to start, as is reviewing the tool selection details presented in the descriptions of expert systems applications in Appendix D.

C.3.7 Filter Available Tools to Identify Candidate Tools

Using the *required* capabilities derived earlier and the discriminating metrics identified, the available tools should be filtered to produce a set of candidate tools to be evaluated in further detail.

C.3.8 Apply the Framework Schema to Evaluate and Select Tools

The final step is the application of the prioritized dimensions to the candidate tools. The appropriate assessment techniques are used to evaluate the relevant metrics applied to each capability of a particular tool in each context, given the relevant application characteristics. Since the dimensions have already been pruned prior to this step, the cross-product of the evaluations performed here will be minimal. A large number of individual evaluations may still be required, but this is necessary. The thoroughness and formality with which these evaluations are made (and with which their weighted results are combined and compared to select a tool or tools) is left to the discretion of the evaluators and end-users.

Appendix D

CASE STUDIES

This appendix contains descriptions of logistics expert system applications listed in alphabetical order by application name. They were collected from the Air Force, Army, Navy, Joint Chiefs of Staff/J4, Defense Systems Management College, and Unisys.

D.1 ACCS, ARMY COMMAND AND CONTROL SYSTEM.

ORGANIZATION. Army Artificial Intelligence Center.
CONTACT. Name: LtC David Tye
Telephone: 202-694-6904, 4141
Address: DACS-DMA
HQDA, OCSA
Room 1D 659
The Pentagon
Washington, D.C. 20310-0200
LOGISTICS AREA. Requirements.
PROBLEM TYPE. Planning.
SHORT DESCRIPTION. ACCS is informally called "planning the war" and was designed to help the Army manage the purchase of weapons worth $13 billion to $20 billion. There are eight new computers and tactical communications systems in battle items of equipment coming into the Army. These new programs have interdependencies, plans for phasing in, and system requirements and issues. The main issue being addressed by ACCS is how the Army should spend its dollars based on detailed information about the eight items of equipment and their effect on over 2500 units. Its knowledge includes: executive guidance, funding status, inter-system dependencies, fielding strategies, materiel requirements, and force structure.
APPLICATION DEVELOPMENT.
Current state of development: this is a prototype system. The total effort to date has been approximately 10 person months: 4 Knowledge Engineers for 2 months and 1 person for 2 months. The effort is ongoing and they plan to add capability at users' requests.
Goal of final development: possibly help others write a Statement of Work for an RFP.
HARDWARE/SOFTWARE REQUIREMENTS. Currently running in KEE on a Symbolics workstation.
SYSTEM INTEGRATION REQUIREMENTS. ACCS must be capable of calls to other languages, must be integrated with a network, and it is critical that it be integrated with databases. It uses an editor, incorporates its own spreadsheet, and is integrated with displays. There are future plans for ACCS to be embedded into a larger information system.
END-USERS.
The intended end-users: ACCS may be used for multiple applications such as logistics, operations, and force structure.
Location of end-users: within Pentagon and could be distributed worldwide.
Estimated number of end-users: 900+ (on mainframe and PCs).
Level of end-user domain expertise: medium.
Level of end-user computing expertise: low.
SYSTEM TESTING PLANS. N/A.
SPECIAL APPLICATION CHARACTERISTICS. ACCS requires arithmetic operators, execution trace, and explaining answers to questions. Direct modifications to the

delivery system are needed. There is a need for the execution module to be separate from the development environment, for the knowledge base to be protected, and for support in rehosting the system to another type of delivery machine. Both backward and forward inference are needed as are conflict resolution, generation of one or multiple or all answers, hypothetical reasoning, inheritance, iteration, pattern matching, simulation, support for other relations, and truth maintenance. Integration with databases is critical and ACCS must be able to make calls to other languages. There is internal access to KEE to set parameter functions. They need model building aids and built their own tools (graphic rule lattice and structure editor) for knowledge-base editing. ACCS requires multiple knowledge bases, will be using rule partitioning in the future, and needs data caching support. They would like intelligent look-ahead and currently do rule compilation. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques include frames, objects, procedures, rules, semantic network, sets, and triggers. Future enhancements of KEE should include the ability to generate automatic interfaces for forms and the specification and use of high level graphics standards.

TOOL SELECTION.

Tools considered: started with OPS5 and YAPS.

Evaluation criteria for tool selection: mainly user interface.

Reasons for tool selected: KEE has a good interface but needs better inferencing techniques.

TRAINING EXPERIENCE. The Army AI Center sent people to KEE classes.

LENGTH OF TIME TO FEEL CONFIDENT. 2 months for those with a strong computer science background and strength in LISP.

VENDOR EVALUATION. After the classes they were dissatisfied with the KEE conflict resolution solution and Intellicorp furnished a special 2-day class to deal with their problems. The KEE documentation is good, but lacking in formal syntax specification. They find it necessary to understand KEE specifications in order to use KEE. Mainly, they have been satisfied with the support, including response to phone calls. KEE reliability and robustness has been good.

OTHER COMMENTS. The Army AI Center staff found the developer and user interfaces to be most critical for the ACCS, especially the ability to generate automatic interfaces for forms.

DOCUMENTS. Selected charts and briefings: AI in the Army, AI Program, Robotics Program Evolution, AI and Robotics Tech Base Groups, Knowledge Engineering Groups, Management Overview (presented at the 1987 Williamsburg Symposium), and ACCS briefing.

LONG DESCRIPTION. ACCS is informally called "planning the war" and was designed to help the Army manage the purchase of weapons worth $13 billion to $20 billion. There are eight new computers and tactical communications systems in battle items of equipment coming into the Army. These new programs have interdependencies, plans for phasing in, and system requirements and issues. The main issue being addressed by ACCS is how the Army should spend its dollars based on detailed information about the eight items of equipment and their effect on over 2500 units. Its knowledge includes: executive guidance, funding status, inter-system dependencies, fielding strategies, materiel requirements, and force structure.

Each system has a force integration staff officer responsible for deciding what units it will be delivered to. Each system has a Department of Army coordinator who addresses the contractor side of the house on how to purchase the system and get it delivered quickly. ACCS needs to represent interdependencies so that if system A is dependent on system B, then A should not have a higher priority than B. For example, once a unit converts to SINCGARS radio it cannot talk to other units with older radios, so it is extremely important to plan which units will receive the new radios in an order that will ensure system integration through communication.

D.2 AFLC FORM 387 REPAIR PROCEDURES ADVISOR.

ORGANIZATION. Air Force Logistics Command/MM-AI.

CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Materiel management.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. Determination of fiscal year equipment item repair requirements is currently accomplished by manual preparation of AFLC Form 387. This method leads to errors in projected requirements and is primarily attributed to inventory manager experience levels and difficulty in determination of quarterly repairable generations. The proposed system will mechanically complete AFLC Form 387 with a minimum of data input by inventory managers. Currently 43 copies are in use for field testing.

APPLICATION DEVELOPMENT.

Current state of development: full scale development. The total effort to date has been 4 person months: 1/2 month each of knowledge engineer, domain expert, and experienced programmer for the exploration stage; the same for the prototyping stage; and 1/2 month Knowledge Engineer and experienced programmer for the development stage.

Goal of final development: fielded system to increase productivity and accuracy.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a PC-compatible micro with 640K RAM.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: inventory managers.

Location of end-users: San Antonio Air Logistics Center Materiel Management.

Estimated number of end-users: 200.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Use database of test cases for consistency checking, verification and validation, and testing changes to evolving system. User feedback is being incorporated during production testing.

SPECIAL APPLICATION CHARACTERISTICS. Domain knowledge was from experience, facts, and procedures. Sources of knowledge were written documentation and people (experts and users). Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are: assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 workshop.

LENGTH OF TIME TO FEEL CONFIDENT. 30 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. Requires knowledge of C to write interfaces which limits use of tool for non-C programmers.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.3 AIRLIFT ALLOCATION.

ORGANIZATION. Joint Chiefs of Staff, Logistics Directorate OJCS/J4.

CONTACT. Name: Don Fowler
Telephone: 202-695-9212
Address: Attn: Major Don Fowler (SCAD)
OJCS, J4
The Pentagon
Washington, D.C. 20301-5000

LOGISTICS AREA. Transportation.

PROBLEM TYPE. Classification.

SHORT DESCRIPTION. Whenever a non-military person wants to fly on U.S. government aircraft they need permission from JCS. It takes an expert almost an entire day to process such a request and it has to be done twice (in theater and at JCS/J4). There are many different categories of people involved such as: ambassadors, spouses, humanitarian missions, disaster relief (e.g., bulldozers for Mexico), negotiating teams, non-DoD U.S. government, non-U.S. government, and foreign nationals. There are several sources offering guidance about how to handle these cases. These include: verbal guidance from the Secretary of Defense, published DoD directives, and Air Force regulations. The objective is to produce an expert system to automate the airlift allocation problem. The user will enter parameters into the system and it will provide an answer with explanation derived from the knowledge base of knowledge integrated from all the guidance sources.

APPLICATION DEVELOPMENT.

Current state of development: initial prototype completed; domain experts are checking the rules and plan to run additional test cases through the system. They will not reimplement the system but will enhance the prototype to reflect the changes. The development has been controlled by J4 but an outside contractor has built the prototype and will complete the fielded system. The prototype required: 2 weeks of Knowledge Engineer, 2 weeks of domain expert, and 3 weeks of AI experienced programmer. Development of the fielded system is expected to take 2 weeks for an AI experienced programmer. The finished system will be approximately 300 rules.

Goal of final development: fielded system that will result in saving of expert's time and better organization of information to avoid ambiguities.

HARDWARE/SOFTWARE REQUIREMENTS. IBM-compatible PC using TIMM.

SYSTEM INTEGRATION REQUIREMENTS. This is a stand-alone system.

END-USERS.

The intended end-users: CINC aids.

Location of end-users: worldwide.

Estimated number of end-users: approximately 12 (1 per CINC).

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium to low.

SYSTEM TESTING PLANS. Inconsistencies of the type where verbal guidance said to do A and written guidance said to do B were caught and resolved by priority. The system does not help discover inconsistencies, the Knowledge Engineer and domain expert have to do so. Verification and validation are being done by collecting a large database of cases. User feedback is being used by the developer to improve the system and there are plans to continue to change, evolve, and enhance the fielded system.

SPECIAL APPLICATION CHARACTERISTICS. The system will perform explanation and some inference, utilize multiple knowledge bases, and use forms, graphics, and text for presentation.

TOOL SELECTION.

Tools considered: only TIMM.

Evaluation criteria for tool selection: mainly experience.

Reasons for tool selected: General Research Corporation is the contractor and they selected their tool, TIMM, because they were experienced in its use.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. Don Fowler is not sold on TIMM because it requires a math co-processor chip and 640K RAM. It was selected as the tool of convenience by the contractor rather than a tool of choice.

OTHER COMMENTS. Don Fowler is an Operations Research Analyst, Specialty Code 49, Action Officer, and has an MS in OR. SCAD is a JCS division for Studies, Concepts and Design. Within SCAD there are three teams: (1) analysis team, (2) systems team, and (3) concepts team. The analysis team does traditional OR type research, mostly mobility, strategic, and some intra-theater mobility analyses. The results of their analyses go to OSD. They do studies and analyses for: JPAM (Joint Program Assessment Memorandum), all service budget review cycle (POMs), RIMS (Revised Inter-theater Mobility Study), DPQ (Defense Planning Questionnaire) centered specifically on NATO, half a dozen or so strategic mobility studies performed on a recurring basis, and others as they come up. There are 4 contractor people supporting them.

DOCUMENTS. Charter for Logistics Artificial Intelligence Coordination Cell (LAICC) (one page).

JCS briefing on Artificial Intelligence (includes a chart on each application).

LONG DESCRIPTION. N/A.

D.4 BAD ACTOR.

ORGANIZATION. Air Force Logistics Command/MM-AI.

CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Maintenance.

PROBLEM TYPE. Diagnosis.

SHORT DESCRIPTION. When an LRU fails on an aircraft but passes the Avionics Intermediate Station (AIS), it becomes known as a "bad actor" and the condition is called Can Not Duplicate (CND). "Bad Actor" guides the user through the many steps required to perform a CND Test Program Software Investigation from beginning to end and provides the user with most of the documentation paths required.

APPLICATION DEVELOPMENT.

Current state of development: full scale development. The total effort has required 7.5 person months. By stage: exploration stage—1 month of Knowledge Engineer, 1/2 month of domain expert, and 1/2 month of experienced programmer; prototype stage—1 month of Knowledge Engineer, 1 month of experienced programmer, and 1/2 month of domain expert; development stage—1 month of Knowledge Engineer and 1 month of experienced programmer; fielding stage—1 month of Knowledge Engineer; maintenance effort is unknown.

Goal of final development: fielded system to increase productivity and accuracy.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Integrates with a DBMS.

END-USERS.

The intended end-users: maintenance and repair personnel.

Location of end-users: Warner Robins Air Logistics Center Maintenance Directorate.

Estimated number of end-users: 15.

Level of end-user domain expertise: high.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. Use database of test cases for consistency checking, verification and validation, and testing changes to evolving system. User feedback was incorporated during prototyping and production testing.

SPECIAL APPLICATION CHARACTERISTICS. Nature of the domain knowledge is: algorithmic, experience, facts numeric, symbolic, and procedural. The application has to handle certainty and uncertainty but doesn't require the consensus of multiple experts. Knowledge sources include: written documentation, people (experts and users), sensor data, and databases. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are: assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 programming.

LENGTH OF TIME TO FEEL CONFIDENT. 30 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. Lack of database interface is delaying the project. A separate product is being acquired to allow non-C programmers to interface M.1 with dBase III.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.5 B-1B CENTRAL INTEGRATED TEST SYSTEM (CITS) EXPERT PARAMETER SYSTEM (CEPS) PROGRAM.

ORGANIZATION. Aeronautical Systems Division/B1 Logistics Resources Engineering Branch (ASD/B1 LRE).

CONTACT. Name: Lt. Sherrie Hegelson
Telephone: 513-255-6528
Address: Lt. Sherrie Hegelson
HQ ASD/B1 LRE
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Maintenance.

PROBLEM TYPE. Diagnosis and monitoring.

SHORT DESCRIPTION. The B-1B Central Integrated Test System (CITS) provides a comprehensive on-aircraft fault detection and isolation capability by on-board monitoring and testing that results in recording over 19,000 parameters. The B-1B CITS Expert Parameter System (CEPS) goes one step further by providing a ground-based maintenance diagnostic tool. Utilizing data recorded on the plane plus aircraft design data, CEPS provides advice for fast, accurate maintenance troubleshooting. CEPS objectives are to improve B-1B diagnostics and to implement the technology required to improve existing and future aircraft diagnostics. CEPS is currently being prototyped for the B-1B offensive avionics, defensive avionics, and airframe subsystems.

APPLICATION DEVELOPMENT.

Current state of development: Phase 2, development of prototypes.

Goal of final development: Phase 1 was a simulation feasibility demonstration from March to September 1985, that ended in a system requirement review and a proposed specification for Phase 2.

Phase 2, the prototype phase, has been under way since February 1986 and has involved three contractors exploring the use of expert system tools for three different B-1B subsystems. The three subsystems (air-frame, offensive avionics, and defensive avionics) are approximately 20% of the B-1B aircraft subsystems but require 80% of the troubleshooting and maintenance efforts.

Phase 3 will be a full scale development and implementation scheduled to begin in January 1988.

The long term goal is the development and maintenance of troubleshooting aids for the entire B1-B aircraft. This will require the development and delivery of fielded expert systems.

HARDWARE/SOFTWARE REQUIREMENTS. The three contractors explored the use of three different expert system tools.

(1) Rockwell addressed the air-frame sub-system using KEE on a Symbolics workstation. The initial prototype effort took 10 person months. The total size of the fielded system is estimated at 2000 KEE units.

(2) Boeing addressed the offensive avionics system using S.1 on a Xerox AI workstation. The initial prototype effort took 10 person months. The total size of the fielded system is estimated at 10,000 S.1 rules.

(3) AIL addressed the defensive avionics using ART on a Sun workstation. The initial prototype effort will take 5 person months. As yet, there is no estimate of the total size of the fielded system.

The current plans for the hardware/software delivery system are to use a Z248 for ASCII graphics, the VAX minicomputer, and probably DEC workstations running S.1.

SYSTEM INTEGRATION REQUIREMENTS. The system will be embedded within a larger information system through a network interface to the Corps Automated Maintenance System (CAMS). The system will be capable of calls to and from other languages. The development contractor will be required to develop software to interface the system with the UNIFY relational database management system. The system will have a built-in word processor and will need to be integrated with a spreadsheet.

END-USERS.

The intended end-users: maintenance technicians.

Location of end-users: 6 geographic locations.

Estimated number of end-users: 1,000.

Level of end-user domain expertise: high to medium, a few at the low end.

Level of end-user computing expertise: low-medium to low.

SYSTEM TESTING PLANS. Consistency checking, system verification and validation, and changing, evolving, and enhancing the fielded system will be done by using a database of test cases. CAMS may be used initially to evaluate how well CEPS performs.

SPECIAL APPLICATION CHARACTERISTICS. The system requires the ability to do arithmetic processing, and will use certainty factors. It requires knowledge-based syntax and semantic checking, and the ability to trace the program execution. The fielded system will be an execution module that is separate from the development environment (i.e., a runtime environment) and the knowledge base must be protected. Inference and control will require: forward and backward chaining, conflict resolution, generation of single or multiple or all answers, hypothetical reasoning, inheritance, iteration, and pattern matching. Graphic rule lattice and structure editors will be needed and there is need for multiple knowledge bases, rule partitioning, and meta-rules that control inferencing. The expert system must be integrable with other software. Optimization by loading several subsystem modules at a time is planned. Presentation will utilize forms, mouse, text, and windows for both end-user and developers and the development environment will require graphics. Representation will include procedures and rules and possibly frames, objects, and a semantic network.

TOOL SELECTION.

Tools considered: KEE, ART, and S.1. S.1 was selected for delivery system.

Evaluation criteria for tool selection: maintainability, speed, and timeliness of decision were the important factors.

Reasons for tool selected: KEE was not selected for two reasons: (1) it is a LISP-based tool and the Air Force software maintainers responsible for maintaining CEPS are not trained in LISP and (2) the performance speed of conventional language-based expert systems was found to be better than that of LISP-based systems. The choice was made between S.1 and ART. Although the ART network approach looked promising, a decision had to be made by summer of 1987 and at that time S.1 with Copernicus was found adequate to do the job. AIL had not had time to do an adequate evaluation of ART and S.1 was selected as the lower risk of the two. Teknowledge, the S.1 vendor, is making changes to extend the tool as requested by the contractor.

TRAINING EXPERIENCE. This information would have to be obtained from the contractors.

LENGTH OF TIME TO FEEL CONFIDENT. This information would have to be obtained from the contractors.

VENDOR EVALUATION. This information would have to be obtained from the contractors.

OTHER COMMENTS. N/A.

DOCUMENTS. Lt. Sherrie Hegelson, "Central Integrated Test System (CITS), Expert Parameter System (CEPS)," in the *Proceedings for the Symposium on Artificial Intelligence for Military Logistics*, March 1987.

Lt. Sherrie Hegelson, "B-1B CITS Expert Parameter System AI Application for Military Logistics," long abstract prepared for the AI Symposium on Artificial Intelligence for Military Logistics, March 1987.

LONG DESCRIPTION. The B-1B Central Integrated Test System (CITS) provides a comprehensive on-aircraft fault detection and isolation capability by on-board monitoring and testing that results in recording over 19,000 parameters. The B-1B CITS Expert Parameter System (CEPS) goes one step further by providing a ground-based maintenance diagnostic tool. Utilizing data recorded on the plane plus aircraft design data, CEPS provides advice for fast, accurate maintenance troubleshooting. CEPS objectives are to improve B-1B diagnostics and to implement the technology required to improve existing and future aircraft diagnostics. CEPS is currently being prototyped for the B-1B offensive avionics, defensive avionics, and airframe systems.

These objectives are being accomplished by applying expert system technology and statistical and data analysis techniques to the recorded B-1B CITS parameters. CEPS consists of three major components: diagnostic documentation, an expert system, and a data analysis system. The components are integrated together with appropriate input/output and control software to perform maintenance assistance functions. The diagnostic documentation provides such items as on-line schematics and signal flow diagrams. This automation provides information to both system engineers and maintenance technicians in concise, usable format.

The expert system is a major portion of CEPS. It supplements the capabilities of both CITS and technical orders (TOs) by providing thousands of unique bits of knowledge associated with aircraft maintenance to the technician. The CEPS Knowledge Engineers gather information from: interviews with design engineers, test engineers and maintenance technicians; design knowledge; maintenance strategy; analysis of CITS parameters; failure and historical data; and maintenance feedback. The information is then incorporated into a commercially available expert system shell. The advantages of the expert system lie in its capabilities to retain information, infer upon this information, provide recommendations based on these inferences, and to "explain" its recommendations.

The data analysis system includes both the database and the analysis tools. The DBMS allows CEPS to view the recorded data and to compare CITS parameter values, as well as to combine, store, sort, and access the data. The analysis tools add the capability to track, trend, and report across all fields within the database in all combinations. This will provide a standard statistical population of 30 flights for each aircraft, without incorporating outdated information. Together these will allow both maintenance technicians and system engineers to examine parameters and specific failures of Shop Replaceable Units (SRUs) or Line Replaceable Units (LRUs) against all historical failures. The values can be compared across one flight, all flights, or a group of flights by location, affiliation, or time. The full scale development CEPS will include two years of on-line CITS data.

CEPS will be incorporated at the organizational and intermediate maintenance levels. Maintenance activities begin with debrief, where CEPS can recall maintenance histories and prompt the debriefer with specific questions. If the failure cannot be duplicated on the ground, CEPS will ask questions and provide advice to assist the maintenance technician. Before LRU repair is initiated, CEPS will provide information on repair times and technical order references. After organizational maintenance is complete, CEPS will record and compare the correct resolution against its initial recommendations. This will provide feedback to CEPS for rapid update. Similar actions occur at the intermediate maintenance level for SRUs. CEPS again collects data to document its performance and facilitate updates to the knowledge base as needed.

The previous system developed for the B-1A bomber defined a large amount of in-flight data to be collected. The current B-1B system differs very much from the original data collection system because the old system was based on the use of conventional computer science techniques and the B-1B system on emerging technology.

Conventional built-in test (BIT) technology deals well with procedurally defined tests but cannot address unique problems that are unanticipated. Expert systems utilizing design engineering information should be able to address unanticipated problems by providing design knowledge about the equipment beyond what is found in maintenance manuals. The experience of senior maintenance experts in the form of heuristics can be included in the expert system, providing an opportunity for their expertise to be used by those with less experience. With CEPS, it is expected that fewer expert maintenance people will be required.

CEPS involves three sub-system prototypes that are being developed by three different contractors: airframe by Rockwell, offensive avionics by Boeing, and defensive avionics by AIL. A fielded prototype of the airframe expert subsystem is expected by the end of August 1987, followed closely by a fielded prototype of the offensive avionics subsystem. The defensive avionics subsystem is currently in the preliminary prototype design phase.

The current system being used on the ground is the CITS Ground Processor (CGP) which is a home-grown file system that sits between the on-board data collection and the Corps Automated Maintenance System (CAMS). CGP has well-defined reports while CEPS is intended to be extensible. Currently the CGP has a one-way link to CAMS but when CEPS replaces the CGP there will be a two-way link between CEPS and CAMS.

CEPS will be justified based on its ability to handle and expedite 20% of the B-1B aircraft subsystems that are responsible for 80% of the troubleshooting and maintenance efforts. CEPS will replace CGP after it has been tested and validated. Currently the on-board data are downloaded to the CGP onto a 9-track tape that is carried to CEPS.

A pre-phase 1 study was conducted to explore the benefit of the approximately 19,000 parameters that were being collected as a part of CITS. An IEEE Spectrum article, "The B-1B CITS Parameter Study" (2 July 1984), reports on this. A result of the parameter study was a suggestion that expert system technology be used. Selection of parameters has been mainly driven by detection and isolation rates and the occurrences of false alarms. Parameters were selected on a case-by-case basis.

Parameter information is collected onboard. The 19,000+ parameters are defined and each has an associated CMC (CITS maintenance code). There are three methods of generating failure reports data.

1. The software continuously checks for failures by seeing if the CMCs are set. If the bit is set then the software will take a snapshot of the parameter, another, 30 seconds later, and a third, 60 seconds later.
2. The crew can hit a switch that will initiate a full parameter snapshot.
3. The system can be set to acquire a complete snapshot sometime during the flight.

CEPS is driven by maintainability, usability, and speed. End-users have been involved in the program from the start as have the life-cycle maintainers of CEPS. It has been planned that Oklahoma City will handle the CEPS maintenance for database interfaces and the operating system. If CEPS fails in the field, it will be rebooted by a logistics command person in the end-user shop.

D.6 CAPITAL INVESTMENT FUNDING CONSULTANT.

ORGANIZATION. Air Force Logistics Command/MM-AI.
CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433
LOGISTICS AREA. Planning/programming resources.
PROBLEM TYPE. Planning.
SHORT DESCRIPTION. This application advises the suitability of project funding considering PRAM, FASCAP, PIF, CSIP. It brings to one source, the regulations and expertise of funding experts on criteria needed to qualify for each funding source. The system guides users through a consultation to help decide which funding source to pursue.
APPLICATION DEVELOPMENT.
Current state of development: fielded system. The total effort has required 1 month of 1 person who served as Knowledge Engineer, domain expert, and programmer. Exploration and prototyping each required .2 month effort, development required .5 month effort and fielding .1 month effort.
Goal of final development: fielded system to increase productivity.
HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.
SYSTEM INTEGRATION REQUIREMENTS. Stand alone.
END-USERS.
The intended end-users: many.
Location of end-users: Ogden Air Logistics Center, all organizations.
Estimated number of end-users: 100.
Level of end-user domain expertise: medium.
Level of end-user computing expertise: medium.
SYSTEM TESTING PLANS. Used test cases for consistency checking and verification and validation. User feedback was incorporated during production testing. The fielded system will be changed, evolved and enhanced by the user. There is no database of test cases.
SPECIAL APPLICATION CHARACTERISTICS. Domain knowledge comes from facts provided by written documents and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are: assumption/rationale history and support for rehosting to another delivery machine.
TOOL SELECTION.
Tools considered: none.
Evaluation criteria for tool selection: used ES tool for rapid development of deliverable system.
Reasons for tool selected: tool provided by a Command-wide site license at low cost.
TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 programming.
LENGTH OF TIME TO FEEL CONFIDENT. 30 days.
VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.
OTHER COMMENTS. N/A.

DOCUMENTS. N/A.
LONG DESCRIPTION. N/A.

D.7 COMPASS, COMMUNICATIONS PLANNING ASSISTANT.

ORGANIZATION. U.S. Army Signal Center, Signal Leadership Dept, C-4 Division, Automatic Branch, ATZH-SLC-A, Fort Gordon.

CONTACT. Name: Captain Rich Routh
Telephone: 404-791-4530
Address: Captain Routh
2528 Springwood Dr.
Augusta, Georgia 30904

LOGISTICS AREA. Not a logistics application.

PROBLEM TYPE. Control, planning, interpretation, execution of implementation, resource allocation, and optimization.

SHORT DESCRIPTION. Assists in communication network planning for brigade and higher echelon levels of the Army. COMPASS is expected to have a significant impact on the usefulness and user acceptance of the Mobile Subscriber Equipment (MSE) that is being bought from the French through GTE and represents considerable modernization.

APPLICATION DEVELOPMENT.

Current state of development: expected completion of fielded system by end of September—all accomplished in-house. The T&E phase of testing will begin in October and will be accomplished by having the users come to Fort Gordon to use the system and critique it. Modifications will be made during this phase. COMPASS has been under development for 2 years and during that time many end-users and experts have contributed to its development. Most were of Lieutenant Colonel or Colonel rank and had attended a 3-week course on AI at Fort Gordon. Part of the course work was to critique COMPASS. There has been constant feedback during development.

There were several stages to COMPASS development. Stage 1 was an exploration stage. During the next stage they: acquired additional staff, began to modify the system to include planning and deployment, incorporated DMA map manipulation data and utilities from CECOM, extended reasoning to include the map data, and finally implemented heuristics for the machine to figure out deployment. Finally, they sent a version to the field.

COMPASS uses 450-500 thousand bytes of memory. It consists of four parts: COMPASS (communication specific); CHARTER (system that controls direct manipulation of graphics by command); utilities system (LISP interacting with the file system); and a color utility system.

Total person years required for prototyping and development are shown below.

Skill Type	Person Years
Design/knowledge acquisition:	
System design/knowledge acquis.	1 genius full time for 2 years
Knowledge Engineer: PhD in AI	1 month over 1 year
System analyst/domain expert	2 people for 6 months during exploration
Managing and marketing to Vice and AI Center	2 months over 1 year
Implementation:	
Domain expert	2 months over first 1.5 years
AI experienced programmer	6 months
Experienced programmer	6 months LISP

Goal of final development: fielded system by January 1988. For field maintenance, they estimate the need for 3 people, 1 full time and 2 part time.

HARDWARE/SOFTWARE REQUIREMENTS. COMPASS is running on a Symbolics workstation with high resolution color graphics. The graphics interface was built by the Army. Rather than using a shell they developed COMPASS in LISP and Flavors because there was no off-the-shelf tool that provided the flexibility and advanced graphics that were needed.

They envision putting COMPASS on a multi-user system (maybe the microvax) with a related database letting the users toggle back and forth to COMPASS as needed. Currently they are not sure there would be enough memory to support multiple users.

SYSTEM INTEGRATION REQUIREMENTS. COMPASS is a stand-alone system that uses DMS map data from tapes loaded into a microvax and then cross loaded to the Symbolics. COMPASS could be integrated with a large screen color display, driving the display from the Symbolics.

END-USERS.

The intended end-users: communication network planners.

Location of end-users: Fort Hood now and later there could be users all over the Army.

Estimated number of end-users: 3-5 now and eventually 100-150.

Level of end-user domain expertise: high.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. They will be able to test the system in practice after the Army gets the mobile subscriber equipment (MSE). There are no formal consistency checks built into the system. Verification and validation of COMPASS was done by personnel at Fort Hood based on an older system (not MSE). COMPASS is continuously changing, evolving, and being enhanced. Fort Hood will probably collect a database of test cases from past exercises and measure the user's capability with COMPASS against exercise data.

SPECIAL APPLICATION CHARACTERISTICS. COMPASS uses arithmetic operators. Explanation uses include explaining user questions, hooks for programmer written explanation, and knowledge-based browsing. The knowledge base is protected and there are plans to rehost the delivery system, possibly on a microvax. Inference techniques used include: conflict resolution, hypothetical reasoning, inheritance, iteration, and support for the transitivity relationship. Knowledge-based editing uses graphic rule lattice and structure editor. Optimization is accomplished through simple rule compilation and search. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques used are: objects and methods, semantic net (the whole problem is analogous to this), triggers and demons in methods.

TOOL SELECTION.

Tools considered: no off-the-shelf tool was suitable.

Evaluation criteria for tool selection: required extensive graphics, flexibility, use of DMA mapping data.

Reasons for tool selected: built in LISP.

TRAINING EXPERIENCE. This information is based on experience in the AI Training Center that Captain Routh runs at Fort Gordon. The AI Training Center has classes in KEE, Picon and M.1. The questions were answered with respect to M.1. Routh feels it is possible to learn the syntax of the tool without a class but not the knowledge methodology and approach. He sent his first instructor off to Teknowledge and is now able to train in-house.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. Documentation that comes with the M.1 course is pretty good but without the course it is not very useful because it does not cover knowledge engineering procedures. He feels Teknowledge has not kept up with the competition in that M.1 does not have a graphics interface or interfaces to application packages such as a DBMS and a spreadsheet. The strength of M.1 lies in being able to build a 2500 rule memory resident system. Shells that are better include: Knowledge Pro (Knowledge Garden), Personal Consultant+, Insight 2+, Nexpert and VPexpert. The M.1 documentation and Teknowledge's response to phone calls and bugs was adequate. The quality of the training is reasonable for the cost. System reliability of M.1 is very good for what it does.

OTHER COMMENTS. N/A.

DOCUMENTS. Information booklet, "The Army's Artificial Intelligence Training Center," 30 June 1987.

LONG DESCRIPTION. N/A.

D.8 CONCEPTUAL DATA MODELER.

ORGANIZATION. Army Artificial Intelligence Center.

CONTACT. Name: Colonel David Tye
Telephone: 202-694-6904, 4141
Address: DACS-DMA
HQDA, OCSA
Room 1D 659
The Pentagon
Washington, D.C. 20310-0200

LOGISTICS AREA. Not a logistics application.

PROBLEM TYPE. Software engineering: data modeling tool.

SHORT DESCRIPTION. Basically this is a database administrator tool that was built to assist in the development of the Army Corporate DataBase (CDB), which has since been suspended because people couldn't agree on terms, data dictionaries, etc. The system aids in building a conceptual data model that helps the user to identify his relations (including many-to-many relations) and build a data dictionary.

APPLICATION DEVELOPMENT.

Current state of development: written in LISP on the Symbolics and currently being ported to KEE.

Goal of final development: N/A.

HARDWARE/SOFTWARE REQUIREMENTS. Runs on the Symbolics workstation and is written in LISP and currently being ported to KEE. The reason for the port is to shift the maintainability and portability issues to Intellicorp. It has been very difficult to keep revising the program for new Symbolics operating system versions.

SYSTEM INTEGRATION REQUIREMENTS. N/A.

END-USERS.

The intended end-users: built for people in Pentagon ISCP for pulling together and modeling Army Corporate DataBase data.

Location of end-users: Pentagon.

Estimated number of end-users: N/A.

Level of end-user domain expertise: N/A.

Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. N/A.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.

Tools considered: KEE because of its availability.

Evaluation criteria for tool selection: availability, cost.

Reasons for tool selected: availability and cost.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Selected charts and briefings: AI in the Army, AI Program, Robotics Program Evolution, AI and Robotics Tech Base Groups, Knowledge Engineering Groups, Management Overview (presented at the 1987 Williamsburg Symposium), and ACCS briefing.

LONG DESCRIPTION. N/A.

D.9 EASES, EXPERT ASSISTANT FOR EQUIPMENT SPECIALIST PROJECT.

ORGANIZATION. University of Southern California/Information Sciences Institute (USC/ISI) under joint Defense Advanced Research Projects Agency (DARPA) and Air Force Logistics Command (AFLC) sponsorship.

CONTACT. Name: Dr. Robert Neches
Telephone: 213-822-1511
Address: Dr. Robert Neches
4676 Admiralty Way
Marina del Rey, California 90292

LOGISTICS AREA. Maintenance/production/depot level/unscheduled repair.

PROBLEM TYPE. Monitoring, prediction, and repair.

SHORT DESCRIPTION. EASES is intended to be an aid to help Equipment Specialists assist in DO41 requirements analysis as members of item management teams at ALCs. It will provide an intelligent support environment for examining data about factors affecting requirements for procurement. The research objectives are to leverage DARPA technology by using a knowledge-based expert system for problem detection, data analysis, solution generation and exploring hypotheticals; to use the knowledge base for help and explanation and to use a high quality user interface to enhance productivity. Research to advance the state of the art in knowledge-based systems includes extending reasoning and representation capabilities, multiple-use knowledge bases, and more sophisticated interplay between the user and the system in performing extended tasks.

APPLICATION DEVELOPMENT.

Current state of development: exploring the use of expert system technology and developing a research prototype. The exploration phase took 1 person year over an elapsed year. The prototype development phase is estimated at a total of 9.5 USC/ISI person years over a 3-year period (2.5 persons during year 1 and 3.5 persons during each of the remaining years) plus 1.5 person years of domain expertise furnished by the Air Force. The technical skills required by person year are shown below:

Skill Type	Exploratory	Prototype
Knowledge Engineer	.25 (AF)	2.5
Domain expert	.25 (AF)	1.5 (AF)
AI experienced programmer	.25	6.0
Manager	.25	1.0

The size of the intelligent workstation development is estimated at 3-5,000 rules for each expert system.

Goal of final development: follow-on to prototype may be RFP for development by contractor or in-house development.

HARDWARE/SOFTWARE REQUIREMENTS. Hardware may be TI Explorer, HP, or Sun workstations. Software tools include NIKL (knowledge representation language and classifier), Qforms (specification language), and Common LISP.

SYSTEM INTEGRATION REQUIREMENTS. The system must be capable of calls to and from DBMS languages such as SQL, must integrate with network and DBMS. EASES is intended to be an integrated part of a larger information system.

END-USERS.

The intended end-users: Air Force equipment specialists.

Location of end-users: located at 12-30 Air Force Bases.

Estimated number of end-users: 1000-1500 users.
Level of end-user domain expertise: medium.
Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. The NIKL classification methodology will provide consistency checking. Verification and validation will be performed by use of a demonstration system that will be delivered at the end of the first year of the project, with new versions appearing at six-month intervals. User feedback will be incorporated into the prototype by ISI. The prototype system will be changed, evolved, and enhanced by the developers at ISI and then delivered to the users in the field. There will be several databases of test cases.

SPECIAL APPLICATION CHARACTERISTICS. Arithmetic operators are needed and supplied by Common LISP. Knowledge-based syntax and semantic checking will be performed by NIKL. The assumption/rationale history and code/data annotation will be done by manual recording. Explanation requires answering user questions, hooks for programmers to provide written explanations, and also hooks for programmers to specify the generation of written explanations.

The development environment utilities will probably be designed to be part of the application. Direct modifications to the delivery system will be performed by Air Force systems support personnel but not by end-users. The execution module is not intended to be separate from the development environment. The knowledge base will be protected from deletion of knowledge but may accept new knowledge. The equipment specialist will only be allowed to change an instance while the Knowledge Engineer may also change representation features.

Types of inference and control include backward and forward chaining, inheritance, pattern matching, and support for relations in addition to inheritance. The tool characteristics will be integrated and the system will be integrated with a DBMS. There will be knowledge-based editing support through use of a graphic rule lattice and structure editors. There will be support for multiple knowledge bases, rule partitioning, and self-organizing data.

Presentation techniques include forms, graphics, mouse, text, and windows. Representation forms include frames, objects, semantic network, and triggers (demons).

TOOL SELECTION.

Tools considered: M.1, S.1, ART, KEE.

Evaluation criteria for tool selection: several criteria were used: (1) classification capability, (2) ability to scale up (final system is expected to be very large), and (3) extensibility.

Reasons for tool selected: NIKL and Common LISP were selected over shells because NIKL provides sophisticated classification capabilities. In terms of hardware, PCs were rejected because of limited disk (there is a need to cache databases), their operating systems did not support the necessary software development tools (e.g., LISP development environment), memory management strategies were not adequate, and the screen was too small.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Ronald Ohlander, "Intelligent Logistics Support Tools," in the *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics*, at Williamsburg, Virginia, March 1987.

Robert Neches, William R. Swartout, and Johanna Moore, "Explainable (and Maintainable) Expert Systems," IJCAI-85, Los Angeles, August 18-23.

Robert Neches, "Tools Help People Co-operate Only to the Extent that They Help Them Share Goals and Terminology," Draft, 1987.

Robert Neches, briefing on EASES, 6/16/87.

Robert Neches, BACKBORD and TINT Demonstration Summary.

LONG DESCRIPTION. EASES is intended to be an aid to help Equipment Specialists assist in DO41 requirements analysis as members of item management teams at ALCs. They will provide an intelligent support environment for examining data about factors affecting requirements for procurement. Research objectives are to leverage DARPA technology by using a knowledge-based expert system for problem detection, data analysis, solution generation, and exploring hypotheticals; to use the knowledge base for help and explanation and to use a high quality user interface to enhance productivity. Research to advance the state of the art in knowledge-based systems includes extending reasoning and representation capabilities, multiple-use knowledge bases, and more sophisticated interplay between the user and the system in performing extended tasks.

General problems being addressed include: (1) the equipment specialist's time is limited and has to be managed as time spent on any item is at the expense of other items; (2) equipment specialists have limited math and statistical skills; and (3) equipment specialists often do not see or pursue important implications and may fail to detect problem indicators buried in the data. Specific support is needed in the areas of: prioritizing items, data verification and analysis, and making judgements involving factor adjustments. Benefits to be gained from EASES include: (1) the use of the notecard facility insures that external information is not forgotten and can be applied to all appropriate items; (2) the specialist's time is concentrated on items that need attention; and (3) explicit audit trails are saved. The NIKL model is used to guide storage and retrieval of user's notes on knowledge outside the bounds of the system and provides a history and audit trail mechanism. User-oriented database browsing can help users find their way in large databases and interface with multiple databases.

Initial accomplishments include: implementation of the initial knowledge base (model of parts, relationships between parts and taxonomy of part-related concepts), model of sections and fields in the DO41 form, representation of data fields and forms for specific items. Also implemented were: rules for detecting questionable data, a notecard facility, and a uniform user interface that supports the same conventions for presenting notecard and DO41 form-manipulation, the beginnings of a calculator facility, and spreadsheet-like management of DO41 data entries. The initial interface is implemented so that each value field on a DO41 form is a NIKL concept. There is a window implementation based on the actual DO41 form. The first year prototype will cover 20 items and support data manipulation and scrubbing. When the equipment specialist corrects depot factors in order to satisfy constraints, the system will keep a history of the change (from depot to base) and give the equipment specialist a chance to enter his reason(s) for the change using the notecard facility.

Main research areas are the architecture and methodologies needed to build support environments and the development of generic tools for end-users as well as for developers. There is a concept of a closed world described by the knowledge in the knowledge base. Knowledge outside of that world may be entered and handled in a special way through notecards. This could include a means for users to enter information that is partially outside of the closed world but references things inside. This kind of information could be in a note that includes references to concepts in the knowledge base.

Planned accomplishments by year:

Year 1: A uniform, high quality interface will present all information needed by the user; an expert system for detecting problem items so that the user will spend time only on items that are problems; and a notecard facility to record outside information and justifications for factor revisions.

Year 2: An expert system to identify bad data and advise on remediation; an expert system to advise on how to adjust factor predictions; and a facility to help users manage time allocated on each item.

Year 3: An expert system to critique user-proposed factor predictions, and a facility to help users explore hypotheticals.

D.10 EPERKS, EXPANDED PERSONNEL REQUIREMENTS KNOWLEDGE SYSTEM.

ORGANIZATION. U.S. Army Logistics Center.
CONTACT. Name: Mr. Oliver Hedgepeth
Telephone: 804-734-1621
Address: Attn: ATCL-OPT (Hedgepeth)
U.S. Army Logistics Center
Ft. Lee, Virginia 23801-6000
LOGISTICS AREA. Requirements.
PROBLEM TYPE. Planning.

SHORT DESCRIPTION. The objective of EPERKS is to aid developers in more rapid and uniform development and modification of the Army Tables of Organization and Equipment (TOEs). Developing the TOE is a complex and time consuming process. The Army Regulations (AR) are difficult to understand and different AR apply to the personnel, equipment, and grade structures. Analysts use their own interpretation of the AR in developing the TOE and thus the TOEs are developed in a non-uniform manner. EPERKS will have a control module and specialized modules for the personnel, equipment, and grade structures. Each specialized module will reflect the rules and procedures from the relevant AR. Functional experts will be consulted so that the AR rules can be augmented by expert "rules of thumb." EPERKS is an expansion of the PERKS prototype.

Use of an expert system is expected to provide more flexibility in the design and development of the TOE, more consistent understanding of the TOE, and better quality control of the product.

APPLICATION DEVELOPMENT.

Current state of development: halfway through design and prototyping. Intellicorp is converting the PERK expert system to a PC host using KEE on the PC as an example for marketing KEE on the PC.

Goal of final development: progress from prototype to RFP to development and fielding of software tool to aid the TOE developer. Size of PERKS is 200 KEE units and estimated size of EPERKS is 2000 KEE units. Person month estimations for PERKS prototype phase and EPERKS development phase are shown below.

Skill Type	PERKS	EPERKS
Knowledge Engineer/system analyst	24	14
Domain expert	6	6
AI experienced programmer		12
Manager	6	2

HARDWARE/SOFTWARE REQUIREMENTS. The prototype is running on a Symbolics 3670 using KEE software. Its development requires access to a PC host and a microvax.

SYSTEM INTEGRATION REQUIREMENTS. EPERKS is capable of calls to LISP and is integrated with a Lotus spreadsheet.

END-USERS.

The intended end-users: developers of PERKS and EPERKS TOE are the Force Development and Evaluation Directorate within the Logistics Directorate. The users are the Quartermaster School and TRADOC schools.

Location of end-users: in U.S.—Fort Gordon, White Sands, all TRADOC schools.

Estimated number of end-users: approximately 200.
Level of end-user domain expertise: medium.
Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Plans are to validate and verify EPERKS using a database of test cases. User feedback is gathered by means of a history file. EPERKS is intended to be a changing, evolving system but the process of supporting this has not been worked out yet. No consistency checking techniques are available or planned.

SPECIAL APPLICATION CHARACTERISTICS. EPERKS needs arithmetic processing. Consistency checking and an assumption/rationale history are needed by the application but are not furnished by KEE. EPERKS will include execution trace, explanation, hooks for programmer written explanation, and knowledge-based browsing. The development environment utilities are not designed to be part of the application and the execution module will be separate from the development environment. Direct modifications to the delivery system will be supported. Protection of the knowledge base is required as is support for rehosting EPERKS to another delivery machine at minimum cost to the user. Inference techniques include forward chaining, generation of single answers only, inheritance, and iteration. The application makes calls to LISP. Knowledge base editing uses graphic rule lattice and structure editor tools. EPERKS requires multiple knowledge bases, partitioned rule sets, and rules controlling inference. Rule compilation is used for optimization. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques include frames, objects, procedures, rules, and semantic net.

TOOL SELECTION.
Tools considered: KEE, OPS5, and LISP (because of familiarity).
Evaluation criteria for tool selection: forward chaining.
Reasons for tool selected: KEE was recommended by Army AI Center.

TRAINING EXPERIENCE. All the LOGCEN expert system programmers have computer science or OR backgrounds with advanced degrees. They are not computer hackers but are interested in sophisticated programming. All training is done within the framework of a project, and though prototypes are developed in-house, AI experts are consulted as needed. (Dr. Feigenbaum has criticized this approach as attacking rather simplistic low payoff problems.)

The LOGCEN sent 2 people for 2 weeks to classes at Symbolics and sent 2 people to classes in Golden Hills Common LISP. Over a 12-month period: 2 people learned KEE and 2 people learned LISP, there was a 3-month in-house tutorial course, and 1 person spent a month at the Army AI Center.

LENGTH OF TIME TO FEEL CONFIDENT. 1 month to be comfortable, 2 years to be competent.

VENDOR EVALUATION. Mixed feelings about Intellicorp. KEE is good but has poor quality control. KEE documents are not kept up-to-date and not well indexed, there were anomalies when converting from KEE version 2 to KEE version 3, and the teachers were not necessarily experts. The best aid was the on-line help, though KEE program examples were good, as were the tutorials. Intellicorp was rated "ok" in response to phone calls and quality of the training for the cost.

OTHER COMMENTS. There is no off-the-shelf shell that solves all the problems but they would prefer to use one to building their own system in a lower level language.

DOCUMENTS. Structure for Army AI Program.

Pat Jones, "Personnel Requirements Knowledge System (PERKS)," in *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics*, Williamsburg, Virginia, 17-19 March 1987.

Oliver Hedgepeth, pamphlet on the Logistics Center, "Logistics AI Projects by Priority."

LONG DESCRIPTION. N/A.

D.11 FAST PARTS PROCUREMENT BROKER AND WORKSTATION.

ORGANIZATION. University of Southern California, Information Sciences Institute (USC/ISI) jointly sponsored by the Defense Advanced Research Project Agency (DARPA) and the Air Force Logistics Command (AFLC).

CONTACT. Name: Dr. Robert Neches
Telephone: 213-822-1511
Address: Dr. Robert Neches
4676 Admiralty Way
Marina del Rey, California 90292

LOGISTICS AREA. Acquisition.

PROBLEM TYPE. Control.

SHORT DESCRIPTION. The rationale of the FAST broker is to provide a simple electronic mail interface to numerous technical data bases and the inventories of major electronics parts vendors. Users will be able to obtain parts easily, quickly, at discount prices, and without large internal overhead charges. FAST is also a testbed for research issues in the areas of intelligent databases, developing alternative government procurement procedures, creating new procurement services to enhance production, and investigation of alternative approaches to message authentication. There are several parts to the system: smart workstation, electronic mail (networking and protocols), and centralized parts broker. The intelligent workstation portion of the system (the expert system related work) is intended to be generic but will be tailored for the FAST users. It is based on previous and continuing research at USC/ISI.

APPLICATION DEVELOPMENT.

Current state of development: the workstation prototype includes (1) knowledge base about specific set of items examinable by general browsing tools, (2) convenient forms-based interface to broker that replaces conventional electronic mail, and (3) preliminary models of procurement scenarios. The knowledge base describes 4,300 different memory chips listed in the "IC Master Handbook," a standard reference volume for finding electronic parts with approximately equivalent specifications.

Goal of final development: N/A.

HARDWARE/SOFTWARE REQUIREMENTS. Prototyping on TI Explorer with plans to port to lower cost HP and SUN workstations.

SYSTEM INTEGRATION REQUIREMENTS. The FAST workstation is part of an integrated system that includes knowledge bases about the local purchasing process and protocols, interface to a conventional electronic mail interface, and interface to the parts broker which includes a database and several knowledge bases.

END-USERS.

The intended end-users: the end-users are intended to be buyers from the Air Force Logistics Command or from the Defense Logistics Agency. A specific set of end-users has not yet been identified.

Location of end-users: possibly the ALC in Sacramento or DLA in Virginia.

Estimated number of end-users: currently unknown.

Level of end-user domain expertise: currently unknown.

Level of end-user computing expertise: expected to be low.

SYSTEM TESTING PLANS. No specific plans yet.

SPECIAL APPLICATION CHARACTERISTICS. Extensive knowledge representation and data browsing techniques.

TOOL SELECTION.

Tools considered: they are developing an integrated toolset. BACKBORD is a browsing aid for knowledge/databases (based on Xerox's RABBIT system) utilizing the NIKL knowledge representation language and classifier that will be interfaced to ORACLE databases. The Scenarios language is used for specifying goals and activities in long-term tasks.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: building own toolset as part of research goals because generic intelligent workstation tools are needed and not available.

TRAINING EXPERIENCE. Researchers were all previously trained on LISP-based workstations.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. The intelligent workstation part of FAST will be based on an Intelligent User Support Environment, which consists of an integrated set of tools to help users perform tasks. The tools include expert systems, user interfaces, NIKL and its successor LOOM which are knowledge base classifiers. (NIKL handles terminological or class knowledge while LOOM is an assertional component.) On the interface side, the approach is to build interfaces that are hand-crafted examples of what is needed, then build custom knowledge bases to support the interfaces and finally develop a general way of combining the interface and knowledge base into a core knowledge base to be used by new applications.

During the discussion and demonstration, it became evident that they are not working directly with any users on the design of the FAST workstation. It would be advantageous to this work if there were buyers (e.g., at ALC or DLA) closely involved in the design and development of the workstation.

Currently USC/ISI has around 15 customers (including the Rome Air Development Center and Schlumberger) and have done between $10,000-15,000 worth of business. Initial problems in setting up USC as the broker had to do with inventory size and liability.

DOCUMENTS. Ronald Ohlander, "Intelligent Logistics Support Tools," in the *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics*, at Williamsburg, Virginia, March 1987.

Robert Neches, William R. Swartout, and Johanna Moore, "Explainable (and Maintainable) Expert Systems," IJCAI-85, Los Angeles, August 18-23.

Robert Neches, "Tools Help People Co-operate Only to the Extent that They Help Them Share Goals and Terminology," Draft, 1987.

Robert Neches, briefing on the FAST Parts Procurement Workstation, 1987.

Robert Neches, BACKBORD and TINT Demonstration Summary.

LONG DESCRIPTION. The rationale of the FAST broker is to provide a simple electronic mail interface for the user to numerous technical data bases and the inventories of major electronics parts vendors. Users will be able to obtain parts easily, quickly, at discount prices, and without large internal overhead charges. FAST is also a testbed for research issues in the areas of intelligent databases, developing alternative government procurement procedures, creating new procurement services to enhance production, and investigation of alternative approaches to message authentication. There are several parts to the system: smart workstation, email, and centralized parts broker.

Streamlined operations are essential to rapid procurement inherent in the FAST concept. Users establish blanket purchase agreements with FAST. FAST places blanket orders with vendors. FAST will only take orders after users have executed agreements which guarantee that FAST will be paid for the parts it buys.

The parts broker serves as a centralized broker for electronic purchase of electronic parts (e.g., "Buy me this part", "Give me quotes on three suppliers of this part", "Buy me the best part according to this quote"). The broker does "global" shopping. Currently, it is a 3-4 person manual operation that could be automated. In order to scale the broker up: (1) the competition advocate has to be convinced that this is a legal way of doing business, and (2) policy considerations have to be worked out that could be based on examples from other procurement regulations. The broker is interposed between the buyer and the vendor and (1) explodes buyer requests to all qualified vendors, and (2) collapses types of interaction between buyer and vendor to requests for quotes and orders. The broker currently does not qualify the vendors.

The intelligent workstation part of the project will eventually supply an intelligent request handler that will help the buyer prepare a correct request. ISI is currently developing a friendly workstation interface to the FAST broker with the capability of customization to support the special needs of individual sites and, in particular, customizations useful to potential DoD users such as ALCs. The project is also developing knowledge bases and knowledge-based reasoning mechanisms which will enhance the functionality of the centralized broker in addition to being useful at local sites.

Analysis of the government purchasing process for certain kinds of items have estimated that approximately 800 decisions must be made during the period counted as administrative lead time. Of these, only a small percentage really require human judgement; the bulk require human action because of an absence of automated information systems. Bottlenecks include the need to determine and follow applicable regulations for each item, tracking the location and progress of paperwork moving through the administrative system, difficulties arising from failures to understand the significance of the items, and a lack of corporate memory that limits the organization's ability to improve performance by reusing previous experience.

Goal of the workstation is to provide a set of capabilities that help alleviate some of these difficulties. These include a friendly interface using advanced graphics and mouse, knowledge about communication protocols for interfacing with the broker, and knowledge about key properties of items being ordered and the purchasing process. Knowledge about communicating with the broker will allow the workstation to advise the user about performing certain tasks (such as requesting price quotes or placing orders). It will use interactive techniques such as pointing, form-filling, and natural language to shield the user from the low-level mechanics of the broker's network interface. The knowledge about key properties of items being ordered at local sites will be used to assist the user in formulation of a purchasing request (for example, by helping the user do research on equivalent parts). Knowledge about purchasing will enable the providing of advice and critiquing of actions with respect to applicable rules and regulations.

A knowledge base suitable for describing a set of about 4300 different kinds of integrated circuits has been constructed and used in conjunction with the BACKBORD (a knowledge-based browser) to demonstrate the capability for users to easily form and refine arbitrary descriptions of parts they are seeking and then be shown all manufacturers satisfying that description. The knowledge base browser acts in conjunction with the knowledge base to enable a user to build descriptions of parts and find items satisfying the description. The browser helps users iteratively refine their queries to better reflect their intent. It lets users see examples retrieved by an initial query, allows them to evaluate whether the results meet their needs, and provides information helpful in deciding how to modify the query. BACKBORD differs from other, related database interfaces in two ways. It provides graphical displays of the information structure in addition to query mechanisms, giving users much more flexibility in gaining understanding of the organization of the information. Further, it is designed to be integrated with other processing activities, rather than merely operating as a stand-alone tool for data inspection.

From a research perspective, the goal of the project is to foster progress in several aspects of AI technology in order to facilitate the construction of intelligent user support environments. Research efforts are proceeding along three major fronts: (1) development of methodology for constructing intelligent support environments (in contrast to a rule-based approach—this emphasizes construction of a common declarative knowledge base usable for multiple purposes); (2) development of generic tools useful in a broad class of situations but designed for customization to particular tasks; and (3) knowledge acquisition and application of techniques.

D.12 FIMS, FORCE INTEGRATION MODELING SYSTEM.

ORGANIZATION. Army Artificial Intelligence Center.
CONTACT. Name: Colonel David Tye
Telephone: 202-694-6904, 4141
Address: DACS-DMA
HQDA, OCSA
Room 1D 659
The Pentagon
Washington, D.C. 20310-0200

LOGISTICS AREA. Requirements.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. FIMS was designed to support General Thurman in answering questions such as: "What are the resource implications for an Army force structure change?" and "What is the impact on Army units if a particular program is cut?" A number of tools were built to support FIMS. They include: (1) Conceptual Data Modeler; (2) the Army Tree Builder (used to see/change the Army organization); and (3) the SSN/LIN knowledge base mapper.

APPLICATION DEVELOPMENT.

Current state of development: this was a 36-man-month prototype effort over 6 months elapsed time. It was developed in LISP on a networked Symbolics. Each knowledge base had different expertise and the various systems worked together. The prototype development was started in KEE but switched to LISP because KEE couldn't handle the data interface and caching problem. There was a need to carefully control the download of large amounts of data into the system and the ability to get rid of it when new data was needed. Since the data being used needs to be in memory, good control and thresholds were needed to get the maximum amount of relevant data loaded. The other expert system shells considered had the same limitation.

Goal of final development: none currently except to decompose FIMS into parts and furnish these to groups that have requested a need for them.

HARDWARE/SOFTWARE REQUIREMENTS. FIMS was running in LISP on a networked Symbolics workstation. It does not run under the latest Symbolics operating system.

SYSTEM INTEGRATION REQUIREMENTS. FIMS requires network access to databases.

END-USERS.

The intended end-users: General Thurman was the original intended user. DCSOPS is interested in the part of FIMS that tracks the force structure and PA&E wants the part that tracks manpower.

Location of end-users: DSCOPS and PA&E in the Pentagon.

Estimated number of end-users: N/A.

Level of end-user domain expertise: N/A.

Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. N/A.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.

Tools considered: KEE and others.

Evaluation criteria for tool selection: interface to database and data caching techniques.

Reasons for tool selected: no tool met the data needs and so FIMS was built in LISP.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Selected charts and briefings: AI in the Army, AI Program, Robotics Program Evolution, AI and Robotics Tech Base Groups, Knowledge Engineering Groups, Management Overview (presented at the 1987 Williamsburg Symposium), and ACCS briefing.

Briefing on Force Integration Model System (FIMS) project.

LONG DESCRIPTION. FIMS was designed to support General Thurman in answering questions such as: "What are the resource implications for an Army force structure change?" and "What is the impact on Army units if a particular program is cut?" A number of tools were built to support FIMS. They include: (1) Conceptual Data Modeler; (2) the Army Tree Builder (used to see/change the Army organization); and (3) the SSN/LIN knowledge base mapper.

The Army planning process begins at the Concepts Analysis Agency. Large analytical models of the threat and Army doctrine are used to arrive at a force skeleton. Essentially, the force skeleton indicates the types and number of units necessary to meet the threats in fighting using the stated doctrine. The DCSOPS develops TOE blueprints of people skills and counts and equipment types and counts for each type of unit. A structured composition is derived by taking the list of units needed from the force skeleton and multiplying each unit by its TOE blueprint. This requirement is then looked at against the available resources to try to apply the resources to satisfy the requirement. Usually, the result is an equipment shortage that generates a procurement requirement that will drive the budget.

The existing planning process tracks 20,000 items of equipment, uses programs and databases that do not communicate with each other and the planning process takes about 1.5 years. It takes 6 months to prepare data for the existing model and 2 weeks to run it. If any changes are made, the entire model has to be run over again.

FIMS currently tracks 50 key items using all kinds of data, e.g., logistics, force structure TOEs, and TOIPs (TOE changes). The TOE blueprint for the unit may change over time and these changes can cause difficulties in force planning. For example, if the type of tank required for the unit changes, then the TOE document is changed and the unit is not considered to be combat ready until the new tanks show up. General Thurman wanted the flexibility to be able to back out the changes and go back to the old TOE, if he desired. FIMS supports this but the old system did not.

FIMS can show the kinds of dollars by year and line item numbers for kinds of equipment. The user can click the mouse on a PDIP dollar item to give a view of the types of equipment or on a line item of equipment to give details about that equipment. The user can get to the specific kind of information that General Thurman had wanted, e.g., those dollars allocated to buy that equipment for these named units, for that PDIP.

Another issue in making Army decisions is what Army is being referenced—there is a need to be able to show/change Army structure rapidly. FIMS uses the Army Tree Builder tool to show the user a MACOM organization structure and lets him change or reorganize the structure rapidly by clicking on units and moving them around. Another tool being used by FIMS is the SSN/LIN knowledge base which was built to map AMC's standard stock number (SSN) into TRADOC's line item number (LIN) and vice versa.

D.13 FIS, FAULT ISOLATION SYSTEM.

ORGANIZATION. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

CONTACT. Name: Dr. Randall P. Shumaker, Director
Kenneth DeJong, Project Leader
Telephone: 202-767-2882, 2884
Address: Naval Research Lab
Navy Center for Applied Research in AI
4555 Overlook Ave., SW
Washington, D.C. 20375-5000

LOGISTICS AREA. Maintenance.

PROBLEM TYPE. Diagnosis.

SHORT DESCRIPTION. FIS was designed primarily to diagnose analog systems, isolating faults to the level of amplifiers, power supplies, and larger components. The methods employed in FIS are also applicable to the automatic generation of the programs that drive conventional automatic test equipment (ATE), to the real-time control of ATE and to fault isolation in systems containing mechanical, hydraulic, optical, and other types of components. FIS assumes the Knowledge Engineer has documentation describing the function and structure of a specific piece of electronic gear called a unit under test (UUT). The documentation includes schematic and block diagrams, specified values of measurable parameters at various test points, and theory of operation. With this documentation, the Knowledge Engineer uses FIS to create a computer model of the UUT. Under the supervision of a technician, FIS later uses the model to recommend tests to make and analyzes the test results until faulty replaceable modules are identified (excerpted from "The FIS Electronic Troubleshooting System," *IEEE Computer Magazine,* July 1986).

APPLICATION DEVELOPMENT.

Current state of development: undergoing testing and debugging on small, primarily analog circuits (e.g., amplifiers, analog multiplexors, and voltage-controlled oscillators) having around 10 modules.

Goal of final development: application of FIS to larger units under test.

HARDWARE/SOFTWARE REQUIREMENTS. Written in Franz LISP and runs on a VAX 11/780.

SYSTEM INTEGRATION REQUIREMENTS. N/A.

END-USERS.

The intended end-users: technicians.

Location of end-users: N/A.

Estimated number of end-users: N/A.

Level of end-user domain expertise: N/A.

Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. Being tested and debugged on small analog circuits.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.

Tools considered: N/A.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: N/A.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Pamphlet on "Navy Center for Applied Research in Artificial Intelligence."

Frank Pipitone, "The FIS Electronics Troubleshooting System," *IEEE Computer Magazine,* Vol. 19, No. 7, July 1986.

LONG DESCRIPTION. FIS was designed primarily to diagnose analog systems, isolating faults to the level of amplifiers, power supplies, and larger components. The methods employed in FIS are also applicable to the automatic generation of the programs that drive conventional automatic test equipment (ATE), to the real-time control of ATE and to fault isolation in systems containing mechanical, hydraulic, optical, and other types of components. FIS assumes the Knowledge Engineer has documentation describing the function and structure of a specific piece of electronic gear called a unit under test (UUT). The documentation includes schematic and block diagrams, specified values of measurable parameters at various test points, and theory of operation. With this documentation, the Knowledge Engineer uses FIS to create a computer model of the UUT. Under the supervision of a technician, FIS later uses the model to recommend tests to make and analyzes the test results until faulty replaceable modules are identified.

The principal goals of the FIS project are to enable FIS to: (1) minimize knowledge acquisition difficulty by equipping it with a library of descriptions of commonly occurring modules and module properties to aid the Knowledge Engineer in producing a concise description of each UUT's connectivity and the function of each UUT's modules (replacement components); (2) compute accurately the probability that a fault diagnosis is correct after one or more tests have been made on a UUT during a diagnostic session; and (3) recommend to the technician the next test to make for maximum information gain and minimum costs in setup changes and measurements. The principal novelties in FIS are the ability to reason qualitatively from a functional model of a complex UUT, without numerical simulation; an efficient knowledge acquisition capability; and a probabilistic reasoning method specialized for device troubleshooting. The basic approach to diagnosis is that of following local causal rules to obtain dynamically all possible causes of various abnormal test results.

In addition to applying FIS to more complex and realistic UUTs, ways to add new capabilities are also under investigation. These include the quantitative relationships among the terminals of a module (e.g., the ratio of an amplifier's output and input RMS voltage is a given gain); automatic deduction of qualitative causal rules from quantitative I/O relations describing modules or the subcomponents of modules; and explanation capability. (This description was excerpted from "The FIS Electronic Troubleshooting System," *IEEE Computer Magazine*, July 1986.)

D.14 FIX, FAULT INVESTIGATION EXPERT PROJECT.

ORGANIZATION. University of Southern California/Information Sciences Institute (USC/ISI) under joint sponsorship by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Logistics Command (AFLC).

CONTACT. Name: Dr. Len Friedman
Telephone: 213-822-1511
Address: Dr. Len Friedman
4676 Admiralty Way
Marina del Rey, California, 90292

LOGISTICS AREA. Maintenance.

PROBLEM TYPE. Diagnosis and repair.

SHORT DESCRIPTION. Throughout the Air Force Logistics Command, depot level diagnosis of avionics equipment relies on a combination of Automated Test Equipment (ATE) hardware and software to detect errors in performance. This is coupled with a massive amount of information provided in Tech Orders (TOs) concerning procedures to troubleshoot a system and remedial actions to take when specific test failures are detected in the ATE. According to the experts interviewed, the TOs, massive as they are, are inadequate as much as 70% of the time. The situation forces diagnosticians to learn special techniques from factory representatives and to devise procedures of their own for effecting repair. FIX is a three-year effort applying advanced artificial intelligence techniques in order to alleviate diagnostic and maintenance problems within the Air Force Logistics Command.

The FIX effort is actually composed of two projects. The near-term applied effort is to build an expert system for a to-be-decided application using a modified version of the Faith tool developed at JPL. The expertise of the domain expert will be captured in a Faith knowledge base and Faith will be used to guide the novice user in diagnosing failures. The second project is a research project whose goal is to develop a new diagnostic tool, FIX, that will learn from experience. The intention is for both projects to address the same application, for the learning system to learn along with the novice, and the results of both systems to be compared to evaluate the FIX expert system shell.

APPLICATION DEVELOPMENT.

Current state of development: the near-term application is based on Faith which is a prototype shell. Faith needs minor modifications in order for it to handle different kinds of notation being used by AFLC to describe circuit diagrams. The long-term FIX learning effort is just getting under way.

An estimate of the number and skills of the people required for the near-term application and the learning system for a 3-year period is presented below.

Year and Skill	Faith Tool	FIX Tool
Year 1		
Technical	1 year	2 years
Domain expert	2-3 months	
Year 2		
Technical	6 months	3.5-4.5 years
Domain expert	2-3 months	
Year 3		
Technical	6 months	3.5-4.5 years
Domain expert	2-3 months	

Faith is currently 6000 lines of LISP code. The knowledge base for a space application took approximately 25 text pages. An estimate of the knowledge base size for future logistics applications is approximately 50 pages.

Goal of final development: an RFP for development by a contractor.

HARDWARE/SOFTWARE REQUIREMENTS. Faith currently runs on the Symbolics and TI Explorer using the Faith expert system shell and Common LISP. Friedman assumes that LISP chips will soon be available on Mac or PC and the contractor who implements the delivery system will do so on that type of workstation. The delivery system should be able to use the knowledge base developed on the Symbolics.

SYSTEM INTEGRATION REQUIREMENTS. Faith and FIX are/will be designed to be stand-alone, unintegrated systems.

END-USERS.

The intended end-users: civilian, novice technicians. The novices generally go through a long learning period (e.g., 18 months) to acquire expertise, and Faith is intended to help them perform better sooner and also to learn while doing the job. The AFLC has systems that need maintenance support for 10 to 15 years and go through many generations of expert technicians. Faith addresses the need for capturing expertise and passing it on. Support of Faith will require civilian software support people with a limited understanding of LISP and the training and ability to do knowledge engineering for Faith. These people are not to be confused with the real end-users.

Location of end-users: all AFLC depots.

Estimated number of end-users: possibly in the hundreds.

Level of end-user domain expertise: medium to low.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Len Friedman has not paid attention to testing the systems developed from Faith but believes that will become important in the production/delivery systems and will be addressed by the implementors. There are currently no plans for validation or verification. User feedback is incorporated into the system by showing the expert how it runs, asking for suggestions for changes and then the system builder alters the system. There will be a database of test cases.

SPECIAL APPLICATION CHARACTERISTICS. An execution trace capability is needed for explanation and knowledge-based browsing will be available when the knowledge base is implemented in NIKL. Inference requires both forward and backward chaining, conflict resolution, hypothetical reasoning, iteration (available with NIKL), simulation, and support for inheritance and other relations. Currently there is no concern with: whether development environment utilities are designed as part of the application; whether direct modifications to the delivery system will be allowed; separation of development and execution models; protection of knowledge base; and rehosting the system. Knowledge acquisition techniques will include rule induction for the learning project. The use of multiple knowledge bases, partitioning of rule sets, rules controlling inference and self organizing rules will enter into the FIX expert tool aspect of the project but not directly to the Faith application. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques include frames, objects, procedures, rules, semantic network, and triggers.

TOOL SELECTION.

Tools considered: none; Faith was developed before there were expert system shells.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: N/A.

TRAINING EXPERIENCE. Faith has only been used by experienced AI people. As an example of training experience, in the past 1 person learned everything about Faith in 4 months and started rewriting Faith code. Friedman expects this to be more difficult for the civilian software system support people because they will not even know LISP when they start.

LENGTH OF TIME TO FEEL CONFIDENT See above.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. There has been some difficulty in finding an application that is important to AFLC and can benefit from the FIX approach and technology.

DOCUMENTS. Ronald Ohlander, "Intelligent Logistics Support Tools," in the *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics,* at Williamsburg, Virginia, March 1987.

Robert Neches, William R. Swartout, and Johanna Moore, "Explainable (and Maintainable) Expert Systems," IJCAI-85, Los Angeles, August 18-23.

Leonard Friedman, "Controlling Production Firing: The FCL Language," in the *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, August 18-23, 1985.

LONG DESCRIPTION. Throughout the Air Force Logistics Command, depot level diagnosis of avionics equipment relies on a combination of Automated Test Equipment (ATE) hardware and software to detect errors in performance. This is coupled with a massive amount of information provided in Tech Orders (TOs) concerning procedures to troubleshoot a system and remedial actions to take when specific test failures are detected in the ATE. According to the experts interviewed, the TOs, massive as they are, are inadequate as much as 70% of the time. The situation forces diagnosticians to learn special techniques from factory representatives and to devise procedures of their own for effecting repair. FIX is a three-year effort applying advanced artificial intelligence techniques in order to alleviate diagnostic and maintenance problems within the Air Force Logistics Command.

The FIX effort is actually composed of two projects. The near-term applied effort is to build an expert system for a to-be-decided application using a modified version of the Faith tool developed at JPL. The expertise of the domain expert will be captured in a Faith knowledge base and Faith will be used to guide the novice user in diagnosing failures. The second project is a research project whose goal is to develop a new diagnostic tool, FIX, that will learn from experience. The intention is for both projects to address the same application, for the learning system to learn along with the novice, and the results of both systems be compared to evaluate the FIX expert system shell.

ISI plans to work with the software support people at McClelland AFB to select an application requiring an automated diagnostician. It is difficult to find the right application because an automated system will only help in routine applications where the majority of what comes in can be diagnosed fairly quickly. The toughest problems are not going to be helped by diagnostic systems. The goal is to find an application in which there is a great deal of routine diagnoses going on and for which the present methods are inadequate and require a lot of training to overcome.

After selecting the application, ISI will build the initial knowledge bases for the application and will train the software support people in using Faith and in building knowledge bases for other Faith applications. Currently, a technician uses the ATE to locate a failure and then consults the TO to debug the failure. The ATE is often very cryptic and may identify a test failure number without additional information; often the TO will have no information about the failure. Faith can follow paths represented in the knowledge base as system block diagrams of the circuits and it can guide the technician in operating like an experienced troubleshooter by consulting the circuit diagram and checking it out in a selective order (i.e., testing paths in the order of their likelihood of failure or if there is no difference then in efficiently partitioning the search space). This effort assumes there is an expert for the application and a new novice that will have to take over the expert's task. Faith differs from other diagnostic system approaches in that it has a built-in general purpose tracing capability.

The FIX learning research approach is to start out with a knowledge base of system block diagrams and circuit diagrams, names in the system, and input-output relationships, and then have the system follow the way in which a novice uses the system to do diagnostics, capturing information about failures into the knowledge base. For example, the system would start out doing a check through the circuit diagrams by a mechanical trace, and then in the future change its trace behavior based on what happened in the past. The assumption is that this will result in minimizing the amount of tracing and eventually will result in spot checking the circuit the way a human expert does, based on what he knows about what fails most often. Block diagrams and circuits are represented in the knowledge base and the system can do traces of the circuits and eventually build up knowledge about what is most likely to fail. With the right kind of models it is believed that explanation-based learning can be supported. The system will learn along with the novice and will become an expert as he becomes an expert. When the next novice comes along, the system will have rewritten itself to furnish expertise to him. The FIX effort will address the same application as Faith so that they can be compared in performance.

D.15 FSA, FORCE SCRUBBER ASSISTANT.

ORGANIZATION. U.S. Army Logistics Center.
CONTACT. Name: Mr. Oliver Hedgepeth
Telephone: 804-734-1621
Address: Attn: ATCL-OPT (Hedgepeth)
U.S. Army Logistics Center
Ft. Lee, Virginia 23801-6000

LOGISTICS AREA. Requirements.

PROBLEM TYPE. Design, resource allocation, pattern matching (examines sequence of numbers and determines if they are in error).

SHORT DESCRIPTION. The Planning Factors Management Division (PFMD) of the Operations Analysis Directorate (OAD) in the Army Logistics Center (LOGC) is the executive agent for the Army logistics planning factors. PFMD maintains a software system that uses biannually made copies of the TOE files at TRADOC Command's Data Processing Field Office. An instance of PFMD work would be producing planning factor data for units in a simulation model. Upon receipt of scenario data in the form of a list of Standard Requirement Codes (an SRC is a code that identifies a single Army unit and its quantities), the requested SRC's are compared against the available SRCs on the PFMD extract. Substitute SRCs are found for those that do not appear on the extract TOE. This is the problem for which AI techniques via an expert system shell were applied. This problem was suggested by a senior analyst, who will be leaving soon, as a means of capturing his expertise.

APPLICATION DEVELOPMENT.

Current state of development: completed. Contains approximately 250 KEE units. Only knowledge source was the expert.

Goal of final development: delivered system.

HARDWARE/SOFTWARE REQUIREMENTS. Runs on Symbolics workstation using KEE shell.

SYSTEM INTEGRATION REQUIREMENTS. Stand-alone system, no integration required.

END-USERS.

The intended end-users: people replacing PFMD expert at Fort Lee.

Location of end-users: Fort Lee.

Estimated number of end-users: 2-3.

Level of end-user domain expertise: low.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Verification and validation were done by comparing the results with a test case "run by hand." User feeds back problems that are fixed on-site. A database of test cases is planned but there is only one test case so far.

SPECIAL APPLICATION CHARACTERISTICS. FSA uses arithmetic operators. Explanation uses include execution trace, explaining user questions, hooks for programmer written explanation, and knowledge-based browsing. Execution module is not separable from the development environment and the development utilities are not intended to be used by the application. Direct modifications to the delivery system are feasible and the knowledge base is protected to some extent. There are no plans to rehost FSA on another delivery machine. Inference techniques used include conflict resolution, forward chaining, inheritance, and pattern matching. Knowledge-based editing uses graphic rule lattice and structure editor. Optimization is accomplished through rule compilation. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques used are frames, objects, semantic net, procedures, and rules.

TOOL SELECTION.

Tools considered: KEE, OPS5 and LISP (because they were familiar with them).

Evaluation criteria for tool selection: forward chaining.

Reasons for tool selected: KEE was recommended by Army AI Center.

TRAINING EXPERIENCE. All the LOGC expert system programmers have computer science or OR backgrounds with advanced degrees. They are not computer hackers but are interested in sophisticated programming. All training is done within the framework of a project, and though prototypes are developed in-house, AI experts are consulted as needed. (Dr. Feigenbaum has criticized this approach as attacking rather simplistic low payoff problems.)

The LOGC sent 2 people for 2 weeks to classes at Symbolics and sent 2 people to learn Golden Hills Common LISP. Over a 12-month period: 2 people learned KEE and 2 people learned LISP, there was a 3-month in-house tutorial course and 1 person spent a month at the Army AI Center.

LENGTH OF TIME TO FEEL CONFIDENT. 1 month to be comfortable, 2 years to be competent.

VENDOR EVALUATION. Mixed feelings about Intellicorp. KEE is good but has poor quality control. KEE documents are not kept up-to-date and not well indexed, there were anomalies when converting from KEE version 2 to KEE version 3, and the teachers were not necessarily experts. The best aid was the on-line help, though KEE program examples were good, as were the tutorials. Intellicorp was rated "ok" in response to phone calls and quality of the training for the cost.

OTHER COMMENTS. There is no off-the-shelf shell that solves all the problems but they would prefer to use one to building their own system in a lower level language.

DOCUMENTS. Structure for Army AI Program.

Jeff Hobbs, "Force Scrubber Assistant (FSA)," in *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics,* Williamsburg, Virginia, 17-19 March 1987.

Oliver Hedgepeth, pamphlet on the Logistics Center, "Logistics AI Projects by Priority."

LONG DESCRIPTION. N/A.

D.16 HEL, HUMAN ENGINEERING LABORATORY EXPERT SYSTEM.

ORGANIZATION. U.S. Army Human Engineering Lab SLCHE-CS.

CONTACT. Name: Richard S. Camden
Telephone: 301-278-5867; Arpanet: CAMDEN@BRL
Address: U.S. Army Human Engineering Lab
Attn: SLCHE-CS
Aberdeen Proving Ground, Maryland 21005-5001

LOGISTICS AREA. Acquisition.

PROBLEM TYPE. Planning, document preparation.

SHORT DESCRIPTION. HEL has defined the Request For Proposal (RFP)/Statement of Work (SOW) product review as their highest priority activity. They plan to develop a knowledge-based system to address the proper insertion of Human Factors Engineering (HFE) text in RFPs. The HFE inputs are intended to be tailored, consistent, legally binding, and complete.

APPLICATION DEVELOPMENT.

Current state of development: their goal was to explore the use of expert system technology and they have developed a preliminary prototype. To date they have: (1) learned about HFE, missile systems, and the procurement process to establish the feasibility of developing an expert system to aid in the generation and tailoring of HFE inputs to RFPs; (2) identified experts and sources of expertise; (3) developed an initial implementation of the prototype expert system; (4) tested the initial implementation with case studies and delivered it to the HEL detachment at MICOM for evaluation; (5) analyzed estimated costs and potential benefits of full system development; and (6) identified alternative approaches and appropriate generic development tools.

Goal of final development: either HEL will write an RFP for development of the prototype by a contractor or the development work will be performed in-house.

They have identified 6 phases in the development of a knowledge-based program within HEL: (1) problem definition—ILIR proposal; (2) prototype development (proposed design for complete system); (3) development of complete system (complete system for MICOM commodity area); (4) iterative refinement (extension to other commodity areas); (5) integration of system (documentation interface with databases, etc.); (6) development of maintenance toolkit (expert-system interface).

Estimated in-house investment requirements for continued development are:

Phases 1 and 2:	time and equipment
Phase 3:	4 man years, $20K software costs
Phase 4:	1 man year/detachment, $10K software costs
Phase 5:	1 man year
Phase 6:	2 man years

HARDWARE/SOFTWARE REQUIREMENTS. Hardware will probably be a high end PC or Mac running NEXPERT or Goldworks.

SYSTEM INTEGRATION REQUIREMENTS. The expert system will require integration with a DBMS and word processor or editing capability and they are unsure as to whether it will be part of a larger information system.

END-USERS.

The intended end-users: writers of RFPs and SOWs.

Location of end-users: N/A.

Estimated number of end-users: N/A.

Level of end-user domain expertise: N/A.

Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. They plan to use consistency checking techniques.

SPECIAL APPLICATION CHARACTERISTICS. The system will require knowledge-based syntax checking and the ability to explain itself to users. Inference requires backward chaining and inheritance and possibly forward chaining. The system will require access to a database. Knowledge-based editing will require a graphic rule lattice and structure editor. Rule partitioning will be needed and eventually there will be a need for multiple knowledge bases. Presentation techniques included forms, graphics, mouse, text and windows. Representation techniques include frames, objects, procedures, rules and semantic network.

TOOL SELECTION.

Tools considered: currently looking at PC and Mac tools.

Evaluation criteria for tool selection: there were several reasons why they feel this is a good expert system application and tools will be evaluated as to these needs. The expert system built from the tool: should be able to backtrack/change users answers to questions; should be easy for an HFE engineer (not a programmer) to maintain; must be capable of incremental development; must have a good user interface; must furnish explanations of its reasoning; must be capable of handling more information, more flexibly and in greater detail than a conventional system; and must handle large knowledge bases in an efficient manner.

Reasons for tool selected: to run on an inexpensive workstation.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. List of AI Tech Base Group Committee members.

List of Combat Service Support Subcommittee—AITBG.

List of AITBG—Subcommittees and Chairpersons.

U.S. Army AI Program/Plan—Coordinated Tech Base (briefing charts).

Draft of Combat Service Support Subcommittee Plan.

Rick Camden's briefing on expert system for RFP generation.

LONG DESCRIPTION. HEL has defined the Request For Proposal (RFP)/Statement of Work (SOW) product review as their highest priority activity. They plan to develop a knowledge-based system to address the proper insertion of Human Factors Engineering (HFE) text in Requests for Proposals (RFPs). The HFE inputs are intended to be tailored, consistent, legally binding and complete.

The knowledge-based system will aid in three areas: information gathering, system evaluation, and document construction. Information gathering is needed to define the document skeleton. The information gathering will use heuristics to: minimize queries to the user by inferring information when possible, query in a user-comfortable order, and gather information in an order intended to reduce the number of subsequent queries.

System evaluation will use heuristics for: default reasoning (e.g., if unknown, then assume best case); offering HFE suggestions about taskings, emphasis areas, DID selection and tailoring, etc.; and checking the completeness and consistency of the document. Document construction will include heuristics for phrase selection, dynamic phrase construction, and phrase structure.

Some examples of rules are:

IF the system is NBC
THEN assume operator needs protective gear.
IF the system is shoulder launched
THEN require a Safe & Army device and add phrase
"an automatic safe and arm firing mechanism is required"
and assume system has shoulder positioner
and assume induction of noise, flash and recoil
and add phrase "shoulder positioner must be stored in a non-obtrusive position."

The pros and cons of continued development have been analyzed. On the positive side, a knowledge-based system is expected to: allow expertise to be perpetuated, ensure a standard quality of response, reduce the required response-time for writing RFPs, result in customized AI tools that could be used for other document generation projects and be used for training purposes. On the negative side: the use of knowledge-based technology may be overkill, it is expected to be costly and time consuming to acquire the necessary knowledge, and acquiring knowledge and building the system will demand time from the most skilled human factors engineers.

D.17 IMA, INVENTORY MANAGER'S ASSISTANT.

ORGANIZATION. Air Force Logistics Command/MM-AI.
CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433
LOGISTICS AREA. Materiel Management.
PROBLEM TYPE. Planning.
SHORT DESCRIPTION. IMA is an ES developed to assist an Inventory Management Specialist (IMS) during the correction cycle of the D041 Recoverable Requirements Computation. D041 is an AFLC data system for determining required inventory levels for recoverable spare parts. IMA helps an IMS validate ten data elements in the D041 system: unit price, date of last procurement, administrative lead-time, production time, base repair cycle time, base processing time, repairable intransit time, supply to maintenance time, shop-flow time, and serviceable turn in time. This ES makes the knowledge and policy of the D041 system more explicit and gives greater accessibility to D041 and IMS expertise.
APPLICATION DEVELOPMENT.
Current state of development: fielded system. The total effort has required 41 person months. By stage: exploration stage—2 months of same person performing knowledge engineering and programming, and 6 months of domain expertise shared among 7+ experts; prototyping stage—6 months of same person performing knowledge engineering and programming, and 14 months of domain expertise shared among 7+ experts; development stage—3 months of knowledge engineering, and 3 months of an experienced programmer; fielding stage—3 months of knowledge engineering, and 3 months of an experienced programmer; maintenance stage—unknown.
Goal of final development: fielded system to increase productivity and accuracy. There are plans to recode IMA in S.1/Copernicus and move it to the IBM mainframe by June 1988.
HARDWARE/SOFTWARE REQUIREMENTS. Requirements are for M.1 running on an IBM compatible PC with 640K RAM. Reasons for the requirements are that the end-user environment has PC's installed and available and this makes it low cost to introduce IMA. The system has been installed on a Zenith Z248 microcomputer costing $3000 running M.1 (unit cost under AFLC site license approximately $1,200).
SYSTEM INTEGRATION REQUIREMENTS. Stand alone.
END-USERS.
The intended end-users: Inventory Management Specialists.
Location of end-users: organizationally the users are in Materiel Management. They are geographically located at:

Warner Robins Air Logistics Center
San Antonio Air Logistics Center
Oklahoma Air Logistics Center
Ogden Air Logistics Center
Sacramento Air Logistics Center

Estimated number of end-users: 1,000.
Level of end-user domain expertise: medium.
Level of end-user computing expertise: low.
SYSTEM TESTING PLANS. Use database of test cases for consistency checking and verification and validation. User feedback is incorporated through deficiency reporting procedures established in the end-user organization. Changes to the fielded system will be handled through an AFLC infrastructure created to support all expert systems.

SPECIAL APPLICATION CHARACTERISTICS. Domain knowledge is algorithmic, heuristic, experiential, factual, symbolic, and both certain and uncertain. Knowledge sources are written documentation, experts, policy managers and databases. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows and rules. Required but not provided by tool are assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: ART, KEE, Personal Consultant, Insight.

Evaluation criteria for tool selection: number of rules, inferencing process, ability to interface with databases, potential to migrate to mainframes.

Reasons for tool selected: "state-of-the-art" capabilities for PC-based system development were provided by the tool selected (M.1).

TRAINING EXPERIENCE. Training supported in-house was 1 week of knowledge engineering and 1 week M.1 programming using vendor's user manuals. Satisfaction with training was moderate to high.

LENGTH OF TIME TO FEEL CONFIDENT. 30 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. The tool selected allows non-programmers who are experts in their domain area to relatively quickly acquire the ability to develop a reasonably robust prototype to demonstrate feasibility of addressing a problem with an expert system. More experienced programmers can advance the prototypes into systems which handle increasingly complex aspects of the problem.

DOCUMENTS. Allen, Mary Kathryn, "The Development of an Artificial Intelligence System for Inventory Management," Council of Logistics Management, Oak Brook, Illinois, 1986.

Allen, Mary Kathryn, and Masters, James M., "The Application of Expert Systems Technology to the Operation of a Large Scale Military Logistics Information System," 1987.

LONG DESCRIPTION. N/A.

D.18 INFORMATION PROCESSING EQUIPMENT FINANCIAL ADVISOR.

ORGANIZATION. Air Force Logistics Command/MM-AI.
CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Information systems.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. The advisor provides funding guidance to personnel ordering information processing equipment or services.

APPLICATION DEVELOPMENT.

Current state of development: full scale development. The total effort has required 1.4 person months. By stage: exploration stage—.2 month of 1 person serving as Knowledge Engineer, domain expert, and programmer; prototype stage—.2 month of 1 person serving as Knowledge Engineer, domain expert, and programmer; and development stage--1 month of 1 person serving as Knowledge Engineer, domain expert, and programmer.

Goal of final development: fielded system to increase productivity.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: maintenance and repair personnel.

Location of end-users: Aerospace Guidance and Meteorology Center, Newark AFB, Ohio (all organizations).

Estimated number of end-users: 100.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Uses test cases for consistency checking and verification and validation; user feedback incorporated during production testing. The system changes will be handled by the users and there is no database of test cases being maintained.

SPECIAL APPLICATION CHARACTERISTICS. The domain knowledge consists of facts and procedures, does not require uncertainty handling nor need consensus of multiple experts. Knowledge sources include written documentation and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 programming.

LENGTH OF TIME TO FEEL CONFIDENT. 30 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. N/A.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.19 JNA, JUSTIFICATION AND APPROVAL DOCUMENT ADVISOR.

ORGANIZATION. Air Force Logistics Command/MM-AI.
CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433
LOGISTICS AREA. Materiel management, contracting/procurement.
PROBLEM TYPE. Planning.
SHORT DESCRIPTION. This ES implements the Competition in Contracting Act (CICA) Justification and Approval Document development process in an ADP environment by automating the legal and policy requirements to produce a draft JNA document from a user friendly consultation session. It can produce a range of JNA files from a blank form to a nearly 100% complete JNA. The program cannot develop or provide all of the unique, valid justification arguments that support a JNA approved format; the user must decide what he/she is going to do, develop the final content and establish why it is being done.

APPLICATION DEVELOPMENT.

Current state of development: full scale development. The total effort has required 7 person months. By stage: exploration stage—1 month of 1 person serving as Knowledge Engineer and programmer, and 1/2 month of a domain expert; prototype stage—2 months of 1 person serving as Knowledge Engineer and programmer, and 1/2 month of a domain expert; development stage—2 months of 1 person serving as Knowledge Engineer and programmer; fielding stage—1 month of 1 person serving as Knowledge Engineer and programmer; and maintenance stage needs are unknown.

Goal of final development: fielded system to increase productivity and quality of end product output. Reason for using an ES tool was to be able to more easily implement the explanatory capability needed.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: all Air Logistics Centers.
Location of end-users:
Warner Robins Air Logistics Center
San Antonio Air Logistics Center
Oklahoma Air Logistics Center
Ogden Air Logistics Center
Sacramento Air Logistics Center.
Estimated number of end-users: 1,000.
Level of end-user domain expertise: medium.
Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Uses test cases for consistency checking and verification and validation; user feedback incorporated during prototype and production testing. There is no database of test cases being maintained.

SPECIAL APPLICATION CHARACTERISTICS. The domain knowledge consists of facts and procedures, does not require uncertainty handling or need consensus of multiple experts. Knowledge sources include written documentation and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are: assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 programming.

LENGTH OF TIME TO FEEL CONFIDENT. 60 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. N/A.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.20 LEARNING CENTER ADVISOR.

ORGANIZATION. Air Force Logistics Command/MM-AI.

CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2555/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Materiel Management.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. This system helps select courses for an individual using the computer learning center in SA-ALC/MM. It asks questions about the individual's computer experience and recommends appropriate lessons. This is a small system which began as an M.1 class project. The system is in use with approximately 26 rules.

APPLICATION DEVELOPMENT.

Current state of development: the system is fielded. The total effort has required 1.2 person months. By stage: exploration stage—1/2 month of 1 person serving as Knowledge Engineer, programmer, and domain expert; prototype stage—1/2 month of 1 person serving as Knowledge Engineer, programmer, and domain expert; development stage—1 month of 1 person serving as Knowledge Engineer and programmer; fielding stage—1/10 month of 1 person serving as Knowledge Engineer and programmer; and maintenance stage needs are unknown.

Goal of final development: fielded system that began as a demonstration project to learn M.1.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: students.

Location of end-users: San Antonio Air Logistics Center, Materiel Management.

Estimated number of end-users: 50.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. Uses test cases for consistency checking and verification and validation; user feedback incorporated as a result of use. There is no database of test cases being maintained.

SPECIAL APPLICATION CHARACTERISTICS. The domain knowledge consists of facts and procedures, does not require uncertainty handling nor need consensus of multiple experts. Knowledge sources include users and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. 1 week of knowledge engineering, 1 week M.1 programming.

LENGTH OF TIME TO FEEL CONFIDENT. 30 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. N/A.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.21 LINDA, WRITING A MODIFICATION PURCHASE REQUEST.

ORGANIZATION. Air Force Logistics Command/MM-AI.

CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Materiel Management, Contracting/Procurement.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. LINDA provides the developers of weapon-systems modification purchase requests guidance on the composition and content of the purchase request and aids in performing a validation check when the purchase request is assembled. The system is named for the expert who contributed significantly to its development. The current system consists of two primary functions, a checklist option and a validate option. LINDA was developed as an IR&D project. Enhancements which would include generation of the final AFLC/AFSC Form 36 are pending.

APPLICATION DEVELOPMENT.

Current state of development: LINDA is a research prototype in field use with enhancements pending. The total effort has required 17 person months. By stage: exploration stage—2 months of a Knowledge Engineer, 1 month of a domain expert, and 1 month of an AI programmer; prototype stage—2 months of a Knowledge Engineer, 1 month of a domain expert, and 2 months of an AI programmer; development stage—3 months of a Knowledge Engineer, 1 month of a domain expert, and 3 months of an AI programmer; fielding stage—1 month of a Knowledge Engineer; and maintenance stage needs are unknown.

Goal of final development: fielded system that began as a demonstration project to learn M.1.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 and C running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: developers of purchase requests for weapon-systems modifications.

Location of end-users: Warner Robins Air Logistics Center, Materiel Management.

Estimated number of end-users: 100.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. Uses test cases for consistency checking and verification and validation; user feedback incorporated as a result of use. There is no database of test cases being maintained. System changes, enhancements will be handled through a future contractor.

SPECIAL APPLICATION CHARACTERISTICS. The domain knowledge consists of facts, experience, and procedures, and does not require uncertainty handling or need consensus of multiple experts. Knowledge sources include written documents, users, and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.
Tools considered: unknown.
Evaluation criteria for tool selection: unknown.
Reasons for tool selected: unknown.
TRAINING EXPERIENCE. Unknown.
LENGTH OF TIME TO FEEL CONFIDENT. Unknown.
VENDOR EVALUATION. Unknown.
OTHER COMMENTS. N/A.
DOCUMENTS. N/A.
LONG DESCRIPTION. N/A.

D.22 LOGDSS, LOGISTICS DECISION SUPPORT SYSTEM.

ORGANIZATION. Joint Chiefs of Staff, Logistics Directorate OJCS/J4.

CONTACT. Name: Don Fowler
Telephone: 202-695-9212
Address: Attn: Major Don Fowler (SCAD)
OJCS, J4
The Pentagon
Washington, D.C. 20301-5000

LOGISTICS AREA. Requirements.

PROBLEM TYPE. Planning, resource allocation, selection (primarily selection/classification).

SHORT DESCRIPTION. The generic problem is to "replace" an ORSA expert in aiding a high level decisionmaker. The prototype application is to do so for supporting JCS/J4 in developing a prioritized critical item list (CIL) that is the result of integrating individual CINC CILs. The generic effort was proposed by an ORSA expert.

APPLICATION DEVELOPMENT.

Current state of development: expert system part of the system is in the conceptual stage. There is a team of 5 people from the University of Kansas led by Dr. Tillman and Dr. Hwong. The level of effort is between 2 and 4 full time equivalents. This is a 2-year project in its second year.

Goal of final development: develop and deliver an expert system as part of a larger system. An additional goal for the JCS/J4 group is the exploration of expert system technology.

HARDWARE/SOFTWARE REQUIREMENTS. IBM PC compatible using Insight 2+.

SYSTEM INTEGRATION REQUIREMENTS. Needs to be capable of calls to/from other languages as Insight 2+ is in Turbo Pascal, and the ORSA algorithms will be written in FORTRAN (Pascal will be the integrating language). Integration with database and displays is necessary, there is a low probability of needing to integrate with a network or word processor. The expert system will be embedded in a larger information system.

END-USERS.

The intended end-users: generic end-users are logistics high level decisionmakers. Prototype end-users are O-6s who support the CINCs.

Location of end-users: worldwide.

Estimated number of end-users: 200.

Level of end-user domain expertise: high.

Level of end-user computing expertise: medium to low.

SYSTEM TESTING PLANS. Plan to do consistency checking. Verification and validation will be done using test cases on a limited scale. The system will be tested against the manual ordering of CILs. User feedback will be passed on to the system developers. There will be a database of test cases (the CIL) and there are plans for the fielded system to continue to change, evolve, and be enhanced.

SPECIAL APPLICATION CHARACTERISTICS. Certainty factors are needed by the application and supported by the tool (the user is asked how sure he is of the answer and bar graphs are used to show percentages).

TOOL SELECTION.

Tools considered: study looked at 40 expert system shells and selected three: M.1, GURU and Insight 2+.

Evaluation criteria for tool selection: listed in study.

Reasons for tool selected: results of study performed for J4 by Transportation Systems Center (TSC). The study was documented but the document has not yet been released.

TRAINING EXPERIENCE. Transportation Systems Center people went to training classes in all three recommended tools (M.1, Insight 2+, and GURU). The people building the system have not gone through training yet. The TSC training experiences will probably be described in the TSC evaluation paper.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. Don Fowler is an Operations Research Analyst, Specialty Code 49, Action Officer and has an MS in OR. SCAD is a JCS division for Studies, Concepts, and Design. Within SCAD there are three teams: (1) analysis team, (2) systems team, and (3) concepts team. The analysis team does traditional OR type research, mostly mobility, strategic, and some intra-theater mobility analyses. The results of their analyses go to OSD. They do studies and analyses for: JPAM (Joint Program Assessment Memorandum), all service budget review cycles (POMs), RIMS (Revised Inter-theater Mobility Study), DPQ (Defense Planning Questionnaire) centered specifically on NATO, half a dozen or so strategic mobility studies performed on a recurring basis, and others as they come up. There are four contractor people supporting them.

DOCUMENTS. Charter for Logistics Artificial Intelligence Coordination Cell (LAICC) (one page).

JCS briefing on Artificial Intelligence (includes a chart on each application).

LONG DESCRIPTION. The generic problem is to "replace" an ORSA expert in aiding a high level decisionmaker. The prototype application is to do so for supporting JCS/J4 in developing a prioritized critical items list (CIL) that is the result of integrating individual CINC CILs. The generic effort was proposed by an ORSA expert.

Essentially, there are many decisions that a Flag Officer or a high ranking decisionmaker has to make, which requires analytical support; and he has to go out and find the right experts in different areas to advise him. The ORSA experts do not have time to serve all decisionmakers and they would like a Decision Support System to replace some of their functions. They wanted a system that has knowledge of the traditional ORSA tools (e.g., proof decisionmaking, multi-attribute decisionmaking, multi-objective decisionmaking), maybe 35 or 40 techniques that are used to solve different kinds of problems. Now, the decisionmaker describes his problem to the expert and the expert decides what kind of technique to use. The LOGDSS expert system would be a front-end embedded in a system to which the decisionmaker would describe his problem. The LOGDSS would evaluate the problem and help the decisionmaker by identifying a possible n out of 40 techniques appropriate to the problem and present the biases and assumptions for each of the suggested techniques. The decisionmaker could filter out techniques on the basis of their assumptions and biases and home in one or two. 10-15% of the system would be devoted to this function. The rest of the system would be devoted to running the selected ORSA algorithm.

This application is being prototyped against J4's CIL problem. Each CINC has to come up with a prioritized critical items list and then J4 has to consolidate the 8 or 9 lists into 1 prioritized list. They need some technique for pulling all of the lists together that has a rational basis in order to get acceptance of their ordering.

The expert system will be a shell to accept a problem description and then figure out, based on parameters and answers to questions, which ORSA techniques are appropriate. About 90% of the ORSA part of the prototype is complete but no work has been done on the expert system yet.

The ORSA people have been working on this for 10-15 years. Dr. Frank Tillman and C.L. Hwang have a consulting group and are doing the work for J4. Frank Tillman is Dean of Mechanical Engineering at Kansas State.

D.23 LOGISTICS DEMAND RATE ANALYSIS EXPERT SYSTEM.

ORGANIZATION. Unisys, Defense Systems, Shipboard and Ground Systems Group, Strategic Systems Division, Logistics Department.

CONTACT. Name: Doug Barton
Telephone: 703-620-1805
Address: Attn: Mr. Doug Bart
Dept. 3682
12010 Sunrise Valley Drive
Reston, Virginia 22091

LOGISTICS AREA: Supply.

PROBLEM TYPE: Prediction.

SHORT DESCRIPTION: Logistics planning and inventory management depend heavily on the accuracy of usage projections for spare and repair parts. Historical data are often used to statistically project future need. Statistical analysis is not entirely reliable when the parts under review have a low operating population and issue rate. Historical data do not supply a sufficient number of data points to produce a reliable regression, and data anomalies can significantly affect the projected demand rate. The problem is further complicated when the parts in question are complex and costly. A number of characteristics of this problem indicate its suitability for an expert system solution. In 1986, a prototype expert system was successfully developed and demonstrated. A production version of this system was in use in August 1987. The system extracts and downloads data from a large logistics database resident on a Unisys 1100 mainframe onto a TI Explorer workstation. A sophisticated set of graphic and tabular tools is provided to allow the logistician to view and manipulate the data. The rule base is implemented in OPS5. The system processes MILSTRIP requisition data and may be adapted to other mainframes and databases.

APPLICATION DEVELOPMENT.

Current state of development: prototype expert system was successfully developed and demonstrated in 1986 and a production version was in use in August 1987. This was supported under Unisys IR&D funding. Labor effort:

Exploratory stage:	1 month of domain expert
	1 month of manager
Prototype stage:	2 months of knowledge engineering and systems analysis
	1 month of domain expert
	2 months of experienced programmer
Development stage:	4 months of knowledge engineering and systems analysis
	2 months of domain expert
	8 months of experienced AI programmer
	2 months of experienced programmer
	2 months of manager

Hardware/software costs were kept to a minimum: the DMS 1100 and its database existed, 1100 Kermit was free, OPS5 was converted from the VAX at no cost, the Explorer configuration cost around $60,000 and the RTMS site license cost $500.

The expert system consists of about 80 rules with more expected.

Goal of final development: to produce and field a production system for Unisys internal use.

HARDWARE/SOFTWARE REQUIREMENTS. Hardware: TI Explorer workstation with a Hayes modem. Software: OPS5, Common LISP, Flavors, TRMS, Kermit.

SYSTEM INTEGRATION REQUIREMENTS. A point-to-point 1250-baud asynchronous line connects the Explorer to the Unisys 1100 mainframe. The expert system uses DBMS data extracted from the DMS1100 DBMS on the 1100 and stores it in the RTMS relational DBMS on the Explorer. The expert system is integrated with a larger information system insofar as it accesses data from the MIS.

END-USERS.

The intended end-users: inventory managers.

Location of end-users: all within the same location and report to the manager of logistics.

Estimated number of end-users: 10.

Level of end-user domain expertise: high.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. Verification and validation were done by using test cases. User feedback is being incorporated and there are plans to change, evolve, and enhance the fielded system. There is no database of test cases and no automated consistency checking.

SPECIAL APPLICATION CHARACTERISTICS. Arithmetic operators are important to the application and handled poorly by the tool. Characteristics important to the application and handled by the tool were: explaining questions, hooks for programmer written explanation, knowledge-based browsing, support for rehosting to another machine, forward chaining, pattern matching, calling other languages, internal access to source code, optimization through rule compilation and rules. Characteristics of importance to the application but not supported by the tool were: assumption/rationale history, consistency checking, execution trace, ability to directly modify the delivery system, ability to separate the execution module from the development environment, protection of the knowledge base, database access, integration of tool characteristics, interprocess calls, internal access to tool parameter setting functions, model building aids, graphic rule lattice, support for multiple databases, rule partitioning, rules controlling inference, presentation techniques (forms, graphics, mouse, text, windows), objects, and procedures.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: none.

Reasons for tool selected: OPS5 was the only tool available at the time on the Explorer and they decided to use it rather than write their own.

TRAINING EXPERIENCE. There were no training classes offered in OPS5 but they did take the Symbolics class to learn the LISP environment and rated that class as superior and a good value. They also used the textbook, *Programming in OPS5*, plus Forgy's users' guide. They rated the Explorer documentation as very good, and the OPS5 documentation as adequate.

LENGTH OF TIME TO FEEL CONFIDENT. 6 to 8 months on the Explorer and 2 months using OPS5.

VENDOR EVALUATION. The version of OPS5 they are using is no longer being supported by the vendor; they have the source code and could extend it but choose not to. The Explorer is fairly stable and reliable but not like a mainframe; OPS5 is very unstable and can crash the Explorer.

OTHER COMMENTS. OPS5 is fast but awkward to work with and has unstable implementation and basically a dead language. The Explorer is a superb, fast prototyping environment once one is familiar with it. It also offers a fine user interface for delivered systems but garbage collection is a problem.

DOCUMENTS. Abstract and briefing charts presented at the Symposium on Artificial Intelligence Applications for Military Logistics.

LONG DESCRIPTION. Logistics planning and inventory management depend heavily on the accuracy of usage projections for spare and repair parts. Historical data are often used to statistically project future need. Statistical analysis is not entirely reliable when the parts under review have a low operating population and issue rate. Historical data do not supply a sufficient number of data points to produce a reliable regression, and data anomalies can significantly affect the projected demand rate. The problem is further complicated when the parts in question are complex and costly. Also, errors are costly.

A number of characteristics of this problem indicate its suitability for an expert system solution. First, there is no reliable statistical solution to the problem. Conventional programming techniques have failed to satisfy the need adequately. Second, the analysis requires access to a large amount of data. Third, the analysis process is procedural in nature, and the evaluation process may be conveniently described by production rules.

In 1986, a prototype expert system was successfully developed and demonstrated. A production version of this system was in use in August 1987. The system extracts and downloads data from a large logistics database resident on a Unisys 1100 mainframe onto a TI Explorer workstation. A sophisticated set of graphic and tabular tools is provided to allow the logistician to view and manipulate the data. The rule base is implemented in OPS5. The system processes MILSTRIP requisition data and may be adapted to other mainframes and databases.

The system is expected to provide more accurate and timely usage projections and reduce the cost to perform the analysis. It will establish a more objective and better-documented evaluation procedure and reduce dependency on "corporate memory." It is also expected to assist in the training of new logistics personnel. A major benefit is better utilization of data stored in conventional mainframe databases. The data are valuable and costly to maintain; the expert system will allow logistics personnel to extract more value from that data.

On completion of the production system, this baseline will be extended to include related logistics disciplines. This work was sponsored by Unisys under their IR&D program.

D.24 MARS, MANAGER'S ASSET RECONCILIATION SYSTEM.

ORGANIZATION. Air Force Logistics Command/MM-AI.

CONTACT. Name: Major Mary Kay Allen and Mr. Paul Dawson
Telephone: 513-257-2655/2657
Address: AFLC/MM-AI
Wright-Patterson AFB, Ohio 45433

LOGISTICS AREA. Materiel Management.

PROBLEM TYPE. Planning.

SHORT DESCRIPTION. MARS is an expert system developed in conjunction with thesis research completed by Capt. Steven McCain while assigned to AFIT. Asset reconciliation is the process of comparing worldwide reported assets with the known asset position and making the two compatible. Failure to properly maintain asset reconciliation records can cause considerable cost increases in inventory and a reduction in mission readiness. McCain's research was conducted to develop an ES and to measure the performance level of the ES in terms of the effectiveness and efficiency of item managers assisted by the system.

APPLICATION DEVELOPMENT.

Current state of development: the system is under full scale development. The total effort has required 10.5 person months. By stage: exploration stage—2 months of 1 person serving as Knowledge Engineer and programmer, and 1/2 month of a domain expert; prototype stage—3 months of 1 person serving as Knowledge Engineer and programmer, and 1/2 month of a domain expert; development stage—2 months of 1 person serving as Knowledge Engineer and programmer, and 1/2 month of a domain expert; fielding stage—2 months of 1 person serving as Knowledge Engineer and programmer; and maintenance stage needs are unknown.

Goal of final development: fielded system to increase productivity and accuracy. Used expert system tool to enable development of explanatory capability more readily than if conventional software had been used.

HARDWARE/SOFTWARE REQUIREMENTS. M.1 running on a Z248 microcomputer.

SYSTEM INTEGRATION REQUIREMENTS. Stand alone.

END-USERS.

The intended end-users: item managers.

Location of end-users: all AFLC Air Logistics Centers.

Estimated number of end-users: 1,000.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: low.

SYSTEM TESTING PLANS. Uses test cases for consistency checking and verification and validation; user feedback incorporated as a result of prototype and production testing. Changes and enhancements to the system will be performed by the users. There is no database of test cases being maintained.

SPECIAL APPLICATION CHARACTERISTICS. The domain knowledge consists of facts, experience, and procedures, and does not require uncertainty handling or need consensus of multiple experts. Knowledge sources include written documents, users, and experts. Important to the application and provided by the tool are: arithmetic operators, knowledge-based syntax checking, code/data annotation, execution trace, explaining questions, execution module separable from development environment, backward chaining, generation of single or multiple or all answers, pattern matching, tool parameter setting functions, rules controlling inference, caching, rule compilation, text, windows, and rules. Required but not provided by tool are assumption/rationale history and support for rehosting to another delivery machine.

TOOL SELECTION.

Tools considered: none.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: tool provided by a Command-wide site license at low cost.

TRAINING EXPERIENCE. Independent study as part of thesis research.

LENGTH OF TIME TO FEEL CONFIDENT. 60 days.

VENDOR EVALUATION. Satisfactory documentation, response to phone calls, training, system reliability/robustness, and program examples.

OTHER COMMENTS. N/A.

DOCUMENTS. N/A.

LONG DESCRIPTION. N/A.

D.25 MOVEMENT PLANNER.

ORGANIZATION. Joint Chiefs of Staff, Logistics Directorate OJCS/J4.
CONTACT. Name: Don Fowler
Telephone: 202-695-9212
Address: Attn: Major Don Fowler (SCAD)
OJCS, J4
The Pentagon
Washington, D.C. 20301-5000

LOGISTICS AREA. Transportation.

PROBLEM TYPE. Planning and resource allocation.

SHORT DESCRIPTION. This application is concerned with planning and re-routing air and sea operations in CENTCOM (Southwest Asia). The objective is to aid in deciding where to land ships. The application uses a map-based representation, mouse interface to the map, and can show different features. It uses a frame based tool where ports have berth slots and berths have slots. The initial prototype will consider a single port. This will be expanded to several ports and finally to the entire area.

APPLICATION DEVELOPMENT.

Current state of development: formulation stage, though they have been trying to get it under way for 2 years. There has been a problem with contracting for this effort. The initial contract is for 2 years and $700K.

Goal of final development: deliver and develop a fielded system. A series of prototypes are planned beginning with a single port, extended to 2 or 3 ports and finally to cover the entire Southwest Asia area. The final system is expected to be large.

HARDWARE/SOFTWARE REQUIREMENTS. Sun/Goldworks.

SYSTEM INTEGRATION REQUIREMENTS. The system will eventually integrate with a network, database, and displays and be capable of calls to/from other languages. Currently it is stand-alone but is expected to become a part of MAPS (i.e., the logistics portion of MAPS is currently not well funded).

END-USERS.

The intended end-users: planners.

Location of end-users: worldwide.

Estimated number of end users: Phase 1 will be three people at 1 CINC; eventually should be extended to 12 people.

Level of end-user domain expertise: high.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. Goldworks supports consistency checking. Verification and validation will be done by running through the actual global exercises. The second prototype is expected to be completed in September 1988 and to run off-line in parallel during an exercise.

SPECIAL APPLICATION CHARACTERISTICS. This application requires arithmetic operators, inference and control includes backward and forward chaining, conflict resolution, generation of single or multiple or all answers, hypothetical reasoning, inheritance, iteration, pattern matching, and truth maintenance. A structure editor is used for knowledge-based editing. The application requires multiple knowledge bases, rule partitioning and rules controlling inference. Optimization if accomplished will occur through rule compilation. Presentation techniques include forms, graphics, mouse, and windows. Representation techniques include frames, objects, procedures, rules, spatial, and possibly semantic network.

TOOL SELECTION.

Tools considered: N/A.

Evaluation criteria for tool selection: N/A.
Reasons for tool selected: N/A.
TRAINING EXPERIENCE. N/A.
LENGTH OF TIME TO FEEL CONFIDENT. N/A.
VENDOR EVALUATION. N/A.
OTHER COMMENTS. Don Fowler is an Operations Research Analyst, Specialty Code 49, Action Officer and has an MS in OR. SCAD is a JCS division for Studies, Concepts and Design. Within SCAD there are three teams: (1) analysis team, (2) systems team, and (3) concepts team. The analysis team does traditional OR type research, mostly mobility, strategic, and some intra-theater mobility analyses. The results of their analyses go to OSD. They do studies and analyses for: JPAM (Joint Program Assessment Memorandum), all service budget review cycle (POMs), RIMS (Revised Inter-theater Mobility Study), DPQ (Defense Planning Questionnaire) centered specifically on NATO, half a dozen or so strategic mobility studies performed on a recurring basis, and others as they come up. There are 4 contractor people supporting them.
DOCUMENTS. Charter for Logistics Artificial Intelligence Coordination Cell (LAICC) (one page).
JCS briefing on Artificial Intelligence (includes a chart on eacn application).
LONG DESCRIPTION. N/A.

D.26 ORGANIZE THE WORLD.

ORGANIZATION. Army Artificial Intelligence Center.
CONTACT. Name: Anthony A. Anconetani
Telephone: 202-694-6904, 4141
Address: DACS-DMA
HQDA, OCSA
Room 1D 659
The Pentagon
Washington, D.C. 20310-0200

LOGISTICS AREA. This is not a logistics application.

PROBLEM TYPE. Design and planning.

SHORT DESCRIPTION. The Organize the World package is in direct support of the reorganization commission for the Army Staff. This package sits on top of the organization data used by the staff and the Office of the Secretariat organization and allows an organization developer to reorganize it or do away with parts of it. There are rules for creating an organization within the Army. For example, one rule says that someone at a particular rank cannot work for someone at the same rank. There are constraints about the size of a subordinate group. The system is able to both enforce and question the constraints. The original objective was to be able to better organize in order to perform the mission. In reality it has been used for understanding and planning reorganization (e.g., when Congress directs the Army to get below a particular force level). It also keeps an audit trail of changes.

APPLICATION DEVELOPMENT.
Current state of development: prototype was built by 2 people in 2 months.
Goal of final development: currently being used at HQ TRADOC.

HARDWARE/SOFTWARE REQUIREMENTS. Symbolics workstation in zetalisp.

SYSTEM INTEGRATION REQUIREMENTS. Stand-alone system.

END-USERS.
The intended end-users: HQ TRADOC.
Location of end-users: HQ TRADOC.
Estimated number of end-users: N/A.
Level of end-user domain expertise: N/A.
Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. N/A.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.
Tools considered: N/A.
Evaluation criteria for tool selection: N/A.
Reasons for tool selected: N/A.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Selected charts and briefings:
- AI in the Army
- AI Program
- Robotics Program Evolution
- AI and Robotics Tech Base Groups
- Knowledge Engineering Group
- Management Overview (presented at the 1987 Williamsburg Symposium)
- ACCS briefing
- Briefing on Organize the World project.

LONG DESCRIPTION. N/A.

D.27 PALOS, PLANNING ASSISTANT FOR LOGISTICS SYSTEMS.

ORGANIZATION. U.S. Army Logistics Center.
CONTACT. Name: Mr. Oliver Hedgepeth
Telephone: 804-734-1621
Address: Attn: ATCL-OPT (Hedgepeth)
U.S. Army Logistics Center
Ft. Lee, Virginia 23801-6000

LOGISTICS AREA. Requirements.

PROBLEM TYPE. Design, planning, prediction, resource allocation.

SHORT DESCRIPTION. The project objective is to adapt the capabilities of the Army Signal Center's Communications Planning Assistant (COMPASS) to the Logistics Center's CSS computer system design and development. The Logistics Automation Directorate (LAD) has responsibility for design and development of CSS computer systems at all echelons for the battlefield to support the supply function. PALOS will be a planning aid for designing the placement of the various computer systems. PALOS will support the user in the manipulation of hardware and software placements as well as the software system interfaces on a battlefield. This will include viewing, modifying, and counting of one or more units at echelons from battalion to corps. COMPASS runs in a development mode on a Symbolics 3640. By mid-August 1987 it will be ready for use by LAD analysts as a tool to visualize multiple system configurations. Still unsettled is the hardware setup at LAD to run the system.

APPLICATION DEVELOPMENT.

Current state of development: development of delivery expert system using COMPASS as a shell. Estimate is 1 person year each of Knowledge Engineer, system analyst, and domain expert to complete system. Size is estimated at 2000 rules.

Goal of final development: delivery system in field.

HARDWARE/SOFTWARE REQUIREMENTS. Symbolics 3640 workstation using COMPASS as a shell.

SYSTEM INTEGRATION REQUIREMENTS. Stand-alone system, no integration requirements.

END-USERS.

The intended end-users: Logistics Automation Directorate.

Location of end-users: Fort Lee but could also be used in Pentagon and DSCLOG.

Estimated number of end-users: 10-20.

Level of end-user domain expertise: high.

Level of end-user computing expertise: medium to high.

SYSTEM TESTING PLANS. No plans for consistency checking and no database of test cases. There are plans to verify and validate the system and incorporate user feedback via a history file. There are plans to change, evolve, and enhance the fielded system.

SPECIAL APPLICATION CHARACTERISTICS. PALOS uses arithmetic operators and may require some form of certainty reasoning. It is concerned with modeling and evaluating distributed and parallel processing systems though PALOS is not itself distributed. It utilizes knowledge-based syntax checking. Development is documented using both assumption/rationale history and code/data annotation. Explanation facilities include execution trace, explanation, hooks for the programmer written explanation, and knowledge-based browsing. The execution module is separable from the development environment. A follow-on activity will be to rehost PALOS to another workstation (e.g., PC). Inference techniques used include backward and forward chaining, conflict resolution, generation of single or multiple or all answers, hypothetical reasoning, inheritance, iteration, pattern matching, simulation, support for other relations and truth maintenance. There is internal access to the source code. Knowledge acquisition includes model building aids. Knowledge-based editing uses graphic rule lattice and structure editor. PALOS uses multiple knowledge bases, rule partitioning, and rules controlling inference. Optimization techniques include intelligent look-ahead and rule compilation. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques include decision tables, images, sensors, frames, objects, procedures, rules, semantic network, sets, and temporal/spatial.

TOOL SELECTION.

Tools considered: KEE, LISP, OPS5.

Evaluation criteria for tool selection: advanced graphics support and flexibility, forward chaining.

Reasons for tool selected: Army AI Center recommendation. COMPASS uses windows and Flavors which allow for the great flexibility that is needed. Symbolics has a superior manipulation graphics user interface that is easily integrated with software.

TRAINING EXPERIENCE. All the LOGCEN expert system programmers have computer science or OR backgrounds with advanced degrees. They are not computer hackers but are interested in sophisticated programming. All training is done within the framework of a project, and though prototypes are developed in-house, AI experts are consulted as needed. (Dr. Feigenbaum has criticized this approach as attacking rather simplistic low payoff problems.)

The LOGCEN sent 2 people for 2 weeks to classes at Symbolics and sent 2 people to learn Golden Hills Common LISP. Over a 12-month period: 2 people learned KEE and 2 people learned LISP, there was a 3-month in-house tutorial course and 1 person spent a month at the Army AI Center.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. Structure for Army AI Program.

Oliver Hedgepeth, pamphlet on the Logistics Center, "Logistics AI Projects by Priority."

LONG DESCRIPTION. N/A.

D.28 PMSS, PROGRAMMER'S SUPPORT SYSTEM KERNEL INTEGRATION MANAGER EXPERT SYSTEM.

ORGANIZATION. Defense Systems Management College.

CONTACT. Name: Mr. Harold Schutt
Telephone: 703-664-4765
Address: Attn: DSMC-DRI-S (Harold Schutt)
Department of Defense
Defense Systems Management College
Ft. Belvoir, Virginia 22060-5426

LOGISTICS AREA. Acquisition.

PROBLEM TYPE. Control, monitoring, planning, and some prediction and repair. Planning and scheduling require determining which activities are the most important and take precedence over others, which varies with the acquisition phase.

SHORT DESCRIPTION. The primary use of the Program Manager's Support System (PMSS) will be in the field as a tool for managers in a program management office. It will assist them in their decisionmaking process and help them execute their project in a more effective and efficient manner. PMSS consists of two major parts, functional modules and the integrated PMSS. Functional modules are software programs that can be used as stand-alone programs to assist in program management areas of responsibility such as planning, acquisition strategy development, program management plan generation, cost estimating, scheduling, budget preparation, etc. These modules support specific functions of program management operations. The integrated PMSS looks across and within all functional areas of responsibility to assess the impact on the program and help the program manager develop alternatives for recovery. The PMSS also provides executive support aids. The portion of the PMSS addressed by this application is expert system support for the PMSS kernel integration manager.

APPLICATION DEVELOPMENT.

Current state of development: the conceptual design has been laid out, the OPS-83 expert system has been ordered, and a demonstrable prototype is scheduled for 1/6/88.

Goal of final development: delivery of a fielded expert system.

HARDWARE/SOFTWARE REQUIREMENTS. Hardware is IBM PC/AT and Zenith 248. They plan to look at 386 machines for version 2. Software: OPS-83 expert system shell, Informix DBMS, and C as the primary implementation language but also BASIC, FORTRAN, and Turbo-Pascal.

Total PMSS development effort is expected to cost $2.4 million over a 4-year period. Approximately $500,000-$700,000 of that will be spent on development of the expert system. Currently the PMSS software programs require a 10M disk and they are telling users to buy 40M disks for the future. No estimate of the size of the expert system portion.

SYSTEM INTEGRATION REQUIREMENTS. The expert system will be an integrated part of the larger information system and its primary purpose is to help in the management of the integrated system. It will need to be capable of calls to/from other languages and integrable with a network, the Informix DBMS, and a word processor/editor.

There are several defined levels of integration:

Level 0: Any module built in BASIC, FORTRAN, or whatever can be dropped into the system and called up from the system menu but without integration; such modules are secondary modules.

Level 1: A level 1 module will use the PMSS user interface but will not use the PMSS database.

Level 2: A level 2 module is called from but does not use the PMSS user interface but it uses the PMSS database.

Level 3: A level 3 module is fully integrated and makes use of the PMSS user interface and PMSS database.

END-USERS.

The intended end-users: members of the acquisition community including those in the Office of the Secretary of Defense, service headquarters, Project Management Offices and field activities.

Location of end-users: worldwide.

Estimated number of end-users: 500-2000.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium to low.

SYSTEM TESTING PLANS. Unknown whether there will be explicit consistency checking. There will be extensive alpha and beta testing; about 20 program management offices have been designated as test sites. DSMC's System X simulation exercise data (based on a real case) will be used for testing. User feedback will be sought and used. The fielded system will be changing, evolving, and enhanced especially during 6-8 months of testing after the prototype demonstration.

SPECIAL APPLICATION CHARACTERISTICS. The application uses arithmetic operators and extended floating point and will utilize parallel processing.

TOOL SELECTION.

Tools considered. besides OPS-83 the tools considered were OPS-5, KES, NEXPERT, Goldworks, S.1, and Expert-Ease.

Evaluation criteria for tool selection: there were no hard technical criteria except that the tool meet the requirements of PMSS, which included the capability to expand for growth. Because potential users number from 500-2000, the cost of the licensing fee was a big consideration.

Reasons for tool selected: N/A.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. DSMC is the only educational institution in DoD dedicated to acquisition management. Its goals are to teach and educate project management personnel in skills needed in management. The primary way this is done is through a 20-week PMC course, which up through 1987 was taught twice a year and starting in 1988 will be taught three times a year. Each class currently is 200 students and will be increased to 270. Much material taught in PMC courses is also taught in short courses given at Fort Belvoir and at 4 regional sites across the country. They teach a total of ~5000 people per year.

Typical PMC student: (1) military student with 14 years of military service, 8 years in acquisition business, and is 38 years old, (2) typical civilian about GS-13 with some number of years in acquisition business. Most students between rank of Captain and 2-star General.

Their job is to teach people acquisition management for the defense acquisition community. The College is organized into three departments: (1) administration, (2) school house, which takes care of the educational side, and (3) the research department. Schutt is in the research department where they developed the idea for the PMSS project in 1981 to help program managers with the information glut—so they could better select, sort, pick the information they needed.

DSMCs funding comes from OSD through the Army. Their research budget is around $1.5 million out of $10 million college operating budget. PMSS has used close to $1 million of the research budget for the last 3 years and will use less in future. PMSS gets support from acquisition activities outside of DMSC.

DOCUMENTS. "The Program Manager's Support System (PMSS), An Executive Overview and System Description," January 1987.

PMSS I Functional Architecture chart taken from the "Red Book" (the new book). The architecture as shown in the document above is outdated.

LONG DESCRIPTION. The primary use of the Program Manager's Support System (PMSS) will be in the field as a tool for managers in a program management office. It will assist them in their decisionmaking process and help them execute their project in a more effective and efficient manner. PMSS consists of two major parts, functional modules and the integrated PMSS. Functional modules are software programs that can be used as stand-alone programs to assist in program management areas of responsibility such as planning, acquisition strategy development, program management plan generation, cost estimating, scheduling, budget preparation, etc. These modules support specific functions of program management operations. The integrated PMSS looks across and within all functional areas of responsibility to assess the impact on the program and help the program manager develop alternatives for recovery. The PMSS also provides executive support aids. The portion of the PMSS addressed by this application is expert system support for the PMSS kernel integration manager.

PMSS provides five basic functions: program impact analysis, program overview-status, functional analysis/support, management aids/tools, and executive support. Program impact analysis is most important; it is the capability to assess rapidly the impact of program perturbations both across and within the areas of interest to the program manager/program management office. The program overview/status function provides the capability to determine easily and quickly the program status in six information categories. The functional analysis/support function is a set of modules that give the program manager additional capabilities related to specific functional areas. Management aids/tools include generic software packages like spreadsheets, word processors, and briefing aids. Executive support functions provide assistance with routine tasks and include capabilities such as calendar, electronic mail, travel plans, etc.

D.29 RAMP, RAPID ACQUISITION OF MANUFACTURED PARTS.

ORGANIZATION. Naval Supply Systems Command.
CONTACT. Name: Mrs. Lorna Estep
Telephone: 202-697-8679
Address: Lorna Estep PML 5505
Commander
Naval Supply Systems Command
Washington, D.C. 20376-5000

LOGISTICS AREA. Acquisition, supply.

PROBLEM TYPE. The RAMP Test and Integration Facility (RTIF) has not yet reached the point in its development of providing a list of system components that are validated candidates for AI technologies. However, the possibilities include:

- Scheduling: OR scheduling generally deals with problems in which all data are known. When data are not known, the problem's complexity can make solutions hard to develop. Moreover, real-world concerns often make OR solutions only ideal approximations to what might actually occur in a factory office environment or on the factory floor. For example, a real-world concern is a part that does not arrive when and where it is supposed to or a worker who calls in sick. An AI model makes dynamic rescheduling possible. The constant monitoring and rescheduling makes it possible to schedule work for individual tools and workstations, to control material delivery to workstations and to track work in progress.
- Process planning: An expert designer-support workstation might be available to preserve knowledge we identify as disappearing, or assist in reverse engineering to develop the process sequence for a part that lacks plans, blueprints, or ready expertise to describe how the part is made.
- Simulation: Simulation can benefit from AI such as SIMKIT software, which provides object-oriented representations and an advanced graphics interface that provides a high productivity environment in which complex simulation models can be built.
- Sensors: RTIF is not expected to deal with high-precision work and so it is possible to consider sensors to monitor cutting tool conditions and compensate for cutting surface wear. Also to compensate for robot inaccuracies, one can install machine vision or tactile feedback sensors to place a robot manipulator in the right place at the right time.
- Robotics: Robotic cells could have knowledge processing components that include elaborate databases and expert systems. Such components would, for example, determine the number of passes each tool might have to make in a sequence of passes, would determine the process sequences among the various tools, would determine the tool cutting speeds, and then would carry out the sequence of passes. Other AI possibilities include planning robot tasks or helping one robot to communicate with other robots. Enabling robots to communicate with each other will make it possible for one robot to machine a part and another to provide further treatment without requiring human intervention.

SHORT DESCRIPTION. The Navy has established RAMP to increase the Fleet readiness by reducing the production (manufacturing and administrative) lead-time for parts that are not available off-the-shelf. This will be accomplished by integrating computer-driven manufacturing technology, part data definition driven production, modular manufacturing workcell architecture, and administrative logistics support to develop the capability to produce large numbers of spare and repair parts on demand. The Navy requires the timely provision of spare and repair parts in order to maintain readiness and provide sustainability. Computer Integrated Manufacturing (CIM), suitably modified, offers the potential for rapid response to fleet requirements. RAMP is a $50 million effort, being run by the South Carolina Research Authority, currently in the conceptual stage and expected to be operational by 1990.

APPLICATION DEVELOPMENT.

Current state of development: the Advanced Logistics Technology Program at NAVSUP in Washington, D.C., has management responsibility for the RAMP project. In 1982, NAVSUP was designated the Lead Systems Command for Navy logistics R&D. Then, in 1983, DoD designated the Navy as the Lead Service for low-volume manufacturing and technical data automation. The RAMP program grew up in the Navy Supply System as a result of these assignments and its development has paralleled similar efforts in other services.

The South Carolina Research Authority has finished the first step by defining the system requirements; it is a leading consortium of the following five companies:

- Arthur D. Little Co: PMO, support engineering
- Battelle Memorial Institute: system engineering, system test and evaluation
- SRI International: printed-wire-assembly center design, development and test
- Ingersoll Engineers: small-mechanical-parts center design, development and test
- Grumman Data Systems: software design, development and test

Other agencies who are working with NAVSUP and NBS on RAMP include Navy laboratories in San Diego, California (NOSC) and White Oak, Maryland; the Naval Supply Center at Charleston (contracting activity for the project); and the Defense Logistics Agency.

Goal of final development: the current schedule for RAMP implementation shows the RTIF in place by mid 1988. The small mechanical parts (SMP) and printed wire assembly (PWA) workcells will be operational in the RTIF by June 1990. The first Navy industrial facility to have an SMP workcell will be the Naval Shipyard at Charleston, South Carolina. This installation will produce some of the parts now being manufactured by using conventional techniques. The first PWA facility will probably be at the Naval Avionics Center, Indianapolis, Indiana.

Two related efforts to RAMP are development of an automated reverse engineering system to develop the technical data required to manufacture parts for which the manufacturer's technical data are not readily available. The second effort is the development of an automated brokering system. This will be a fully computerized system, probably installed at the Inventory Control Point level, which would analyze each requirement for a replacement or replenishment part and determine the best source, whether that be issue from stock, provision of a substitute item from stock, procurement from conventional commercial sources, or manufacture in a RAMP facility. If an order is placed under RAMP, the automated brokering system would select a vendor based on qualification and workload, establish priorities among competing orders in the same facility, and handle all fiscal control procedures. Such a system would permit computerized order inquiries and/or changes throughout the manufacturing process.

HARDWARE/SOFTWARE REQUIREMENTS. N/A.

SYSTEM INTEGRATION REQUIREMENTS. N/A.

END-USERS.

The intended end-users: N/A.

Location of end-users: N/A.

Estimated number of end-users: N/A.

Level of end-user domain expertise: N/A.

Level of end-user computing expertise: N/A.

SYSTEM TESTING PLANS. N/A.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.

Tools considered: N/A.

Evaluation criteria for tool selection: N/A.
Reasons for tool selected: N/A.
TRAINING EXPERIENCE. N/A.
LENGTH OF TIME TO FEEL CONFIDENT. N/A.
VENDOR EVALUATION. N/A.
OTHER COMMENTS. N/A.

DOCUMENTS. Stefan Shrier, "Rapid Acquisition of Manufactured Parts (RAMP) Test and Integration Facility," in *Proceedings for the Symposium on Artificial Intelligence Applications for Military Logistics*, Williamsburg, Virginia, March 1987.

"RAMP Breaks Ground in Charleston," from *The Navy Supply Corps Newsletter*, May/June 1987.

Stefan Shrier, briefing on RAMP presented at the Symposium on Artificial Intelligence Applications for Military Logistics at Williamsburg, Virginia, March 1987.

"RAMP: Improving Logistics Support Through Technology," short descriptive paper given to us by Lorna Estep.

LONG DESCRIPTION. The Navy has established RAMP to increase the Fleet readiness by reducing the production (manufacturing and administrative) lead-time for parts that are not available off-the-shelf. This will be accomplished by integrating computer-driven manufacturing technology, part data definition driven production, modular manufacturing workcell architecture, and administrative logistics support to develop the capability to produce large numbers of spare and repair parts on demand. The Navy requires the timely provision of spare and repair parts in order to maintain readiness and provide sustainability. Computer Integrated Manufacturing (CIM), suitably modified, offers the potential for rapid response to fleet requirements. The Navy has established RAMP to pursue this goal.

The Navy has a need to obtain vital small parts that are commercially unavailable in the U.S.—either because they are too expensive or require far too much time to manufacture. With RAMP, the Navy expects to get a 90% reduction in lead-time for obtaining these special parts, both mechanical and electrical.

RAMP is expected to be operational by 1990. It is a $50 million effort, currently in the conceptual phase and is being run by the South Carolina Research Authority. Lorna Estep expects to hold a preliminary design review in August 1988 and a detailed design review in January 1988. The reviews will be followed by the next stage—developing specifications for procurement. This will require a study of areas in which off-the-shelf AI technology could be used.

The objective of the RAMP program is to develop the capability to produce selected classes of out-of-stock or out-of-production parts on demand. Under the RAMP concept, computer interpretable specifications for a required item will be communicated to an automated manufacturing facility and the part will be produced in a flexible manufacturing workcell and shipped directly to the end-user.

A flexible manufacturing system (FMS), as defined by the Department of Commerce (DOC), "includes at least three elements: a number of workstations, an automated material handling system, and system supervisory computer control." A flexible manufacturing cell, according to DOC, "generally has more than one machine tool with some form of pallet changing equipment such as an industrial robot or other specialized material handling device." The workcell is more narrowly defined because its mission is more specific.

The thrust of the RAMP program is the development of prototype flexible manufacturing workcells to produce small mechanical parts (SMP) and printed wiring assemblies (PWA). An SMP is defined as a mechanical part that will fit into a 1-foot cube. A PWA is an electronic assembly consisting of a printed circuit board with electronic components installed on it.

By the end of 1990, there should be a RAMP installation at the Naval Shipyard in Charleston that is concerned with small parts and integrated with a parts ordering system that uses electronic ordering. This may use the USC/ISI FAST program for ordering electronic components, i.e., Westinghouse may use FAST to get component parts to integrate into larger assemblies. This could involve FAST using the bill of materials for selected boards. FAST could select distributors who would supply parts at the best price or according to other selected attributes.

D.30 STOCKPOINT INVENTORY MANAGEMENT EXPERT SYSTEM.

ORGANIZATION. Operations Analysis and Material Requirements Determination Division (SUP 042) within Inventory and Informations Systems Development Directorate (SUP 04).

CONTACT. Name: Dr. Robert D. Schultz, SUP 042A
Telephone: 202-695-6879, 202-695-6865
Address: Dr. R.D. Schultz SUP 042A
Naval Supply Systems Command
CM#3, Rm 801
Washington, D.C. 20376-5000

LOGISTICS AREA. Supply.

PROBLEM TYPE. Interpretation and prediction.

SHORT DESCRIPTION. There are eight Naval Supply Centers that have inventory managers (IMs) who establish, obtain, and sustain levels for a wide range of inventory items. The goal of this project is to develop an expert system to raise the quality of item management through training new item managers and by providing a systematic and consistent procedure for all item managers. An expert system module at each Inventory Control Point (ICP) will illustrate how an IM could interact with the Uniform Inventory Control Points System (UICP) to determine wholesale inventory requirements for consumables and repairables and provide an auditable trail with statistics for management review.

APPLICATION DEVELOPMENT.

Current state of development: a conceptual prototype was developed during a Phase I Small Business Innovation Research (SBIR) project. Currently a contract proposal for the Phase II SBIR effort is being readied for submittal. An operational test of the Dues Management Advisor is the first required option under the Phase II SBIR project. Given management approval of the operational test results, other options can be activated to develop advisory capabilities in replenishment review, item range maintenance, demand deviation analysis, and updating of primary item data files.

The prototype effort took 4-1/2 person years of Knowledge Engineer and system analyst skills over a 24-month period.

Goal of final development: tested and validated full-scale prototype.

HARDWARE/SOFTWARE REQUIREMENTS. Zenith 240/M.1 and integration with Tandem computer.

SYSTEM INTEGRATION REQUIREMENTS. Integration with network, database, and word processor. The system will be embedded in a larger information system.

END-USERS.

The intended end-users: item managers.

Location of end-users: eight Naval Supply Centers Code 101: Norfolk, Charleston, Jacksonville, Pensacola, San Diego, Oakland, Puget Sound, and Pearl Harbor. Potential use at other Navy stock points.

Estimated number of end-users: 250-300.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium.

SYSTEM TESTING PLANS. No decision has been made as to how to implement consistency checks but the system will incorporate user feedback and there will be support for changes and enhancements to the fielded system.

SPECIAL APPLICATION CHARACTERISTICS. Include support for uncertainty and consensus of multiple experts.

TOOL SELECTION.

Tools considered: considered GURU in addition to M.1. Decided to use expert system because (1) they have reached the limit in use of analytic models and (2) there were continuing "soft spots" in field reviews.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: contractor has used M.1 and made the selection.

TRAINING EXPERIENCE. This has included the contractor personnel attending classes, using documentation, on-line help, tutorials, and program examples. The satisfaction has apparently been high.

LENGTH OF TIME TO FEEL CONFIDENT. Consult contractor.

VENDOR EVALUATION. Consult contractor.

OTHER COMMENTS. They have two small business programs under way. The Stockpoint Inventory Management system is being done by Dialog Systems using M.1. The other project is being done by Digital Development Corp. using GURU. The contractors will be cooperating in Phase II.

They began a new system design effort called "re-solicitation" about 10 to 15 years ago as a two-phased approach: transition phase and re-systemization phase. The transition phase will phase out old hardware and system software bringing application software up to Standard COBOL 76 standardization (minimal changes). The re-systemization phase is relooking into UICP functional needs and implementation as represented by 72 functional descriptions (FD) and has gone on for the past 6-7 years. Some of the FDs are still being completed. They are now having small business contractors examine whether expert system technology is applicable to the supply and demand function.

How did they select the supply and demand function? They have responsibility for ICPs and stock control points, and knew that supply and demand had great variability and expert systems looked like a new theory to apply. Some driving reasons for using the new technology was the recognition that they have experts, some of whom will be retiring, and realization of the great variability in people making decisions. Also the supply/demand review is done every two weeks now, and in the future they expect to do it every day.

Mr. George Bernstein discussed a set of issues/concerns about expert system technology. He believes that no one in the DoD will take expert systems seriously until the private sector shows that they can work. His concerns include (1) how to introduce expert system technology in a smart way that will be successful, (2) the need for a clearinghouse of successful applications (no failures), (3) the need for consolidated acquisition support for buying expert system packages and for soliciting contractor support through RFPs, and (4) the need for consolidated training, i.e., a place to send people to be trained at a reasonable cost.

DOCUMENTS. "FMSO News, Notes and Quotes," a short note on Navy Stock Fund Allotments.

"SBIR#38 Demonstration Plan," writeup on the Stockpoint Inventory Management application.

AI Functional Categories for Logistics and examples of those. This also includes a preliminary list of AI programs RAMP will research.

LONG DESCRIPTION. N/A.

D.31 TEMP, TEST AND EVALUATION MASTER PLAN AUDITOR.

ORGANIZATION. Defense Systems Management College.
CONTACT. Name: Mr. Harold Schutt
Telephone: 703-664-4795
Address: Attn: DSMC-DRI-S (Harold Schutt)
Department of Defense
Defense Systems Management College
Ft. Belvoir, Virginia 22060-5426

LOGISTICS AREA. Acquisition.

PROBLEM TYPE. Monitoring, planning, repair, and evaluation of TEMPs.

SHORT DESCRIPTION. The TEMP Auditor module is an expert system to aid a manager in a program management office; an action officer in a component Acquisition Command or Headquarters; and action office in the Deputy Under Secretary of Defense (Test and Evaluation) Office; or the Office of the Director of Operational Test and Evaluation. The module helps the user evaluate a Test and Evaluation Plan after the TEMP has been written. This module development was funded by the Office of the Secretary of Defense.

APPLICATION DEVELOPMENT.

Current state of development: the contract is officially completed but the prototype was inadequate and will need additional work by George Washington University staff. It is believed the problem was with GWU management in keeping the development team together and getting the job done. The total cost to date has been $160,000 over a 15-month period. The staff included knowledge engineering and system analysis skills and a domain expert from the government. The system currently runs off a 360K disc and is approximately 100 rules.

Goal of final development: develop and deliver a fielded expert system.

HARDWARE/SOFTWARE REQUIREMENTS. IBM PC XT using M.1. The total cost to date has been $160,000 over a 15-month period.

SYSTEM INTEGRATION REQUIREMENTS. This is a stand-alone system.

END-USERS.

The intended end-users: Program Managers and Action Officers.

Location of end-users: OSD, test-and-evaluation organizations of the service.

Estimated number of end-users: 500 PMs and 20 Action Officers.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: medium to low.

SYSTEM TESTING PLANS. When this project is finished, it will be tested at field sites. It should incorporate user feedback and the fielded system is intended to be changing, evolving, and enhanced. A database of test cases would come from a real program.

SPECIAL APPLICATION CHARACTERISTICS. N/A.

TOOL SELECTION.

Tools considered: selected by contractor.

Evaluation criteria for tool selection: N/A.

Reasons for tool selected: selected as part of technical proposal.

TRAINING EXPERIENCE. N/A.

LENGTH OF TIME TO FEEL CONFIDENT. N/A.

VENDOR EVALUATION. N/A.

OTHER COMMENTS. N/A.

DOCUMENTS. "The Program Manager's Support System (PMSS), An Executive Overview and System Description," January 1987.

LONG DESCRIPTION. N/A.

D.32 VAXMGR, VAX MANAGER'S ASSISTANT.

ORGANIZATION. U.S. Army Logistics Center.
CONTACT. Name: Mr. Oliver Hedgepeth
Telephone: 804-734-1621
Address: Attn: ATCL-OPT (Hedgepeth)
U.S. Army Logistics Center
Ft. Lee, Virginia 23801-6000

LOGISTICS AREA. Maintenance.

PROBLEM TYPE. Diagnosis, instruction, interpretation, repair.

SHORT DESCRIPTION. The VAXMGR is an expert system for guiding users through problem identification and resolution by using VAX management and experience expertise as well as VAX documentation. Many of the problems resolved by the VAXMGR are recurring and could be solved by experienced VAX users given appropriate guidance. The VAXMGR could be applicable to all VAX VMS systems within the Army being of benefit to system users in the absence of a manager. It also serves as a training tool for new VAX assistant managers. The system should reside on hardware other than the VAX in order to be available when the VAX system is done.

APPLICATION DEVELOPMENT.

Current state of development: completed and being used at Fort Lee Logistics Center. It has 60 rules and was built by 1 programmer in 1 month partly during the M.1 training class.

Goal of final development: enhancements to the VAXMGR. Future plans are to design and build a manager for the IBM mainframe.

HARDWARE/SOFTWARE REQUIREMENTS. Runs on a PC using M.1 expert system shell.

SYSTEM INTEGRATION REQUIREMENTS. Stand-alone system, no integration.

END-USERS.

The intended end-users: VAX users.

Location of end-users: Fort Lee now and plan to expand to Fort Gordon and Fort Leavenworth.

Estimated number of end-users: 10 as of August 1987, plan to expand to 40-50 by end of year.

Level of end-user domain expertise: medium.

Level of end-user computing expertise: high.

SYSTEM TESTING PLANS. VAXMGR was validated and verified using a database of test cases. User feedback is gathered by means of a history file. No consistency checking techniques are available or planned.

SPECIAL APPLICATION CHARACTERISTICS. VAXMGR uses confidence factors. Explanation includes execution trace, explanation of user's questions, hooks for programmer written explanation and knowledge-based browsing. The development utilities are intended to be part of the delivery system though the execution module is separable from the development environment. Direct modifications to the delivery system are not supported and the knowledge base is protected. The system can be rehosted on other PCs. Inference techniques used are backward chaining, conflict resolution, inheritance, and iteration. Knowledge-based editing tools used are graphic rule lattice and structure editor. Presentation techniques include forms, graphics, mouse, text, and windows. Representation techniques used are procedures and rules.

TOOL SELECTION.

Tools considered: only M.1.

Evaluation criteria for tool selection. N/A.

Reasons for tool selected: M.1 was in-house.

TRAINING EXPERIENCE. M.1 was in-house for a 30-day practice period before the programmer attended a 2-week M.1 training class. VAXMGR development started in the M.1 class.

LENGTH OF TIME TO FEEL CONFIDENT. By the end of the 2-week class, the programmer felt confident.

VENDOR EVALUATION. Found Teknowledge to be a responsive and good vendor—they were responsive to phone calls and fixing bugs. M.1 reliability was good. On-line help, tutorials, and program examples were all helpful and the documentation was good.

OTHER COMMENTS. N/A.

DOCUMENTS. Structure for Army AI Program.

Oliver Hedgepeth, pamphlet on the Logistics Center, "Logistics AI Projects by Priority."

LONG DESCRIPTION. N/A.

REFERENCES

[Boehm 1988] Barry W. Boehm, "A Spiral Model of Software Development and Enhancement," *Computer*, Vol. 21, No. 5, May 1988, pp. 61–72.

[Brooks 1987] Frederick P. Brooks, Jr., "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer*, Vol. 20, No. 4, April 1987, pp. 10–19.

[Kolodziej 1988] Stan Kolodziej, "The Fate of 4GLs," *ComputerWorld*, February 3, 1988, pp. 25–28.

[Rothenberg 1987] Jeff Rothenberg, Jody Paul, Iris Kameny, James R. Kipps, and Marcy Swenson, *Evaluating Expert System Tools: A Framework and Methodology—Workshops,* N-2603-DARPA, The RAND Corporation, July 1987.

[Tracz 1988] Will Tracz, "Confessions of a Used-Program Salesman—Programming-in-the-New," *Computer*, Vol. 21, No. 5, May 1988, p. 74.