NAVAL POSTGRADUATE SCHOOL
Monterey, California

DTIC
ELECTE
JUN 2 2 1989
S E D

THESIS

EXPERT SYSTEMS
AND
COMMAND, CONTROL, AND COMMUNICATION
SYSTEM ACQUISITION

by

JAMES E. MINNEMA

March 1989

Thesis Advisor:	E. N. HART
Co-Advisor	C. R. JONES

89   6 20 037

Unclassified

Security Classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings

2a Security Classification Authority	3 Distribution Availability of Report
2b Declassification/Downgrading Schedule	Approved for public release; distribution is unlimited.

4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)

6a Name of Performing Organization Naval Postgraduate School	6b Office Symbol (If Applicable) 74	7a Name of Monitoring Organization Naval Postgraduate School

6c Address (city, state, and ZIP code) Monterey, CA 93943-5000	7b Address (city, state, and ZIP code) Monterey, CA 93943-5000

8a Name of Funding/Sponsoring Organization	8b Office Symbol (If Applicable)	9 Procurement Instrument Identification Number

8c Address (city, state, and ZIP code)	10 Source of Funding Numbers			
	Program Element Number	Project No	Task No	Work Unit Accession No

11 Title (Include Security Classification) Expert Systems and Command, Control, and Communicaton System Acquisition

12 Personal Author(s) James E. Minnema

13a Type of Report Master's Thesis	13b Time Covered From To	14 Date of Report (year, month,day) March 1989	15 Page Count 103

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (continue on reverse if necessary and identify by block number)
Field	Group	Subgroup	Expert systems, Acquisition management, Decision Support Systems, M.1

19 Abstract (continue on reverse if necessary and identify by block number)

This thesis examines the organizational causes of the Department of Defense's (DoD) inability to acquire working defense systems. One major cause of this is identified as a lack of a sufficient number of trained and experienced acquisition personnel. An examination of the definitions of Decision Support and Expert Systems is made to determine their suitability for application to this problem. The information system framework of Gorry and Scott Morton is used to structure the acquisition problem. The DoD acquisition problem is found to be a good candidate for the application of expert systems. An expert system architecture is developed to provide acquisition personnel both technical and management support. Use of a central mainframe, connected to the Defense Data Network will provide nationwide access, with centralized control of the knowledge base. Thie architecture allows for the incorporation of existing conventional software under expert software control. In order to reduce development cost and time, the use of existing DoD manuals, as the knowledge base, is proposed. A prototype module, utilizing the M.1 expert shell and DoD Manual 4245.7-M and NAVSO P-6071 is developed to prove the feasibility of this approach.

20 Distribution/Availability of Abstract [X] unclassified/unlimited [ ] same as report [ ] DTIC users	21 Abstract Security Classification Unclassified

22a Name of Responsible Individual LCDR E. Neil HART	22b Telephone (Include Area code) (408) 646-2966	22c Office Symbol 54HR

DD FORM 1473, 84 MAR   83 APR edition may be used until exhausted   security classification of this page

All other editions are obsolete   Unclassified

EXPERT SYSTEMS
AND
COMMAND, CONTROL, AND COMMUNICATION
SYSTEM ACQUISITION

by

James E. Minnema
Captain, United States Marine Corps
B. S., Purdue University, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(COMMAND CONTROL AND COMMUNICATIONS)

from the

NAVAL POSTGRADUATE SCHOOL
March 1989

Author: _____
James E. Minnema

Approved By: _____
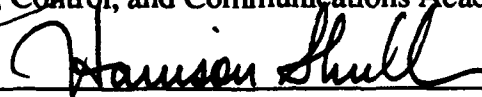E. Neil Hart, Thesis Advisor

_____
Carl R. Jones, Co-Advisor

_____
Carl R. Jones, Chairman
Command, Control, and Communications Academic Group

_____
Harrison Shull
Provost and Academic Dean

ii

ABSTRACT

This thesis examines the organizational causes of the Department of Defense's (DoD) inability to acquire working defense systems. One major cause of this is identified as a lack of a sufficient number of trained and experienced acquisition personnel. An examination of the definitions of Decision Support and Expert Systems is made to determine their suitability for application to this problem. The information system framework of Gorry and Scott Morton is used to structure the acquisition problem. The DoD acquisition problem is found to be a good candidate for the application of Expert Systems.

An expert system architecture is developed to provide acquisition personnel both technical and management support. Use of a central mainframe, connected to the Defense Data Network will provide nationwide access, with centralized control of the knowledge base. The architecture allows for the incorporation of existing conventional software under expert software control. In order to reduce development cost and time, the use of existing DoD manuals, as the knowledge base, is proposed. A prototype module, utilizing the M.1 expert shell and DoD Manual 4245.7-M and NAVSO P-6071 is developed to prove the feasibility of this approach.

For
&I
id
tion

Distribution/
Availability Codes

Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I. INTRODUCTION

In 1985, the Washington Post ran a series of articles, titled Defense INC., illustrating some of the problems occurring in the Department of Defense (DoD) acquisition system (Washington Post 1985). These problems range from excessive requirements, increased Congressional oversight, and low maintainability, to excessive profits, and unethical conduct in contracting. In 1985, Secretary Weinberger was forced to cancel the DIVision Air Defense (DIVAD) air defense system after it failed operational testing (Smith 1988:172). In 1988, a scandal erupted involving the alleged bribery of DoD procurement officials for insider information. Since its inception and Presidential announcement, the Strategic Defense Initiative (SDI) program has been embroiled in controversy over its feasibility and workability (Smith 1988:603-616). Lastly, these problems and others were perceived to be so bad, and the DoD so unable or unwilling to fix them, that the Congress stepped in and mandated changes in the assignment and training of program management personnel (President's Blue Ribbon Commission on Defense Management 1986:28).

These examples illustrate that the DoD is experiencing problems in trying to develop and procure the complex weapon systems it needs in the numbers and time frame dictated by

1

modern technology. This is not to say that these problems are necessarily new or unique to the 80s. Indeed, defense fraud has been around since the Revolution, and will be around as long as people and profit are part of the procurement process. However, the above headlines do suggest that a look needs to be taken at the reasons why the recent scandals have occurred and why it takes ten years or more to develop a new weapon system that often does not work as advertised (U.S. Congress, Senate 1986:566).

What makes the DoDs problems so serious is that the United States (US) is entering an era of limited resources, both fiscal and industrial, and waste denies critical amounts of material to the defense forces of the US. Furthermore, current challenges to US industrial and nuclear supremacy, mean that any weakening of the US defense capability can not easily be made up. In light of decreased US industrial capacity, it is imperative that DoD procurements minimize their drain on the national economy while not weakening the defense capabilities of the US (U.S. Congress, Senate 1986:553). In order to accomplish this, it is necessary to correct the DoD procurement process so that it works more efficiently and will therefore require less resources, both capital, labor, and material.

Another important aspect of US defense capability, is to improve our ability to use the existing forces in the inventory. This is the area of Command, Control, and

Communications (C3). There is a growing awareness that C3 can be either a significant force multiplier or divider (Herres 1983:31). This means that the proper C3 can allow a weaker force to prevail against superior forces and conversely poor C3 can prevent a strong force from completing its mission. Therefore, one way the US can reduce the drain of defense on the national economy is to possess effective C3 systems.

However, the DoDs problems with procurement also affect the procurement of C3 systems. This leads to C3 systems that are developed in time, operate poorly, and contribute to a lack of effective C3, thereby reducing the effectiveness of existing US forces. Furthermore, the field of C3 is very dependent on fast changing computer technology. Yet this technology is one of the most difficult to incorporate into weapons systems. Therefore, in order to increase the effectiveness of existing US forces, it is critical that C3 systems be developed quickly and work as planned.

In order to allow the efficient procurement of weapon systems, and to allow the procurement of effective C3 systems, it is necessary to determine the causes for the DoDs inability to acquire working defense systems. Only after the causes are determined is it possible to determine if a solution can be found. The purpose of this thesis is to take a careful look at several of the potential

3

organizational causes of this problem. These causes will then be examined to determine if it is possible to utilize expert computer systems to assist acquisition personnel in managing their complex and difficult jobs.

II. PROBLEM AND THEORETICAL BASIS FOR SOLUTION

A. PROBLEM DISCUSSION

A brief description of the Program Manager's (PM's) task begins this discussion. There have been many attempts to describe the PM's job, each with varying degrees of succinctness and clarity. The problem is that the PMs job deals with every aspect of the project, and definitions try to include every aspect. One of the best definitions that the author has found, comes from the Navy Program Manager's Guide 1987 (Draft). It states the following description of the PM's responsibility:

> PMs, within their chartered responsibility, shall exercise technical and business/financial management for the accomplishment of the program objectives within approved constraints and thresholds. In order to do this, the PM will need to develop a broad array of managerial skills. Many of these skills will have their locus in the program management organization and support activities, but certain ones must reside in the PM himself.
>
> The PM will be the primary advocate for the program. At the outset, the prospective PM must be thoroughly convinced of the need which the program addresses before he takes on the PM responsibility. He must completely understand the military need for the system and must become intimately familiar with the system as it evolves. Since a series of minor decisions can have a major impact on the program, the PM must understand and appreciate the implications of each trade-off decision. (U.S. Department of the Navy 1987:2-1)

What this passage is pointing out, without mentioning them specifically, is that the PM must be aware of all the fields of knowledge relating to the design, manufacture, and

production of a weapons system. This means he or she must manage the use of high technology components, design theory, application, etc. Often the PM is not an expert in any of these fields. Regardless of this fact, the Manual goes on to make the most important point about the PM's role: "The PM must understand that he and he alone is responsible and accountable for the success or failure of the program." (U.S. Department of the Navy 1987:2-1).

In order to accomplish this monumental task, and shoulder this responsibility, the program manager is given a number of personnel for assistance. With these personnel he or she must form a management team that is capable of performing the above task description. These personnel vary in nature from civilian and military personnel to private support contractors. Furthermore, the technical, managerial, and program management backgrounds of these individuals varies; with the PM being dependent upon the military personnel system, the availability of a civilian staff, and the expense of hiring qualified support contractors. It is with these personnel resources that the PM must form an effective management team.

Unfortunately, in the past there has been considerable variance in the expertise and ability of the personnel assigned to the PM job and his or her staff. In support of this criticism, in 1985 a staff report to the Senate Armed Services Committee highlighted this as one of their points

for improvement (U.S. Congress, Senate 1986:560). And at the same time the President's Blue Ribbon Commission on Defense Management reported the following:

...The defense acquisition work force mingles civilian and military expertise in numerous disciplines for management and staffing of the world's largest procurement organization. Each year billions of dollars are spent more or less efficiently, based on the competence and experience of these personnel. Yet, compared to its industry counterparts, this work force is undertrained, underpaid, and inexperienced. Whatever other changes may be made, it is vitally important to enhance the quality of the defense acquisition work force - both by attracting new personnel and by improving the training and motivation of current personnel.

...We also support recent legislation that has further defined career paths for all program managers. In 1984, Congress established a minimum four-year tenure for program management assignments. The 1986 Authorization Act prescribed requisite qualifications and training, including at least eight years of acquisition-related experience and appropriate instruction at the Defense Systems Management College (or equivalent training).

By contrast, much more remains to be done concerning civilian acquisition personnel generally. Civilians frequently cite the rigid pay grades and seniority-based promotion standards of the federal civil service as disincentives to continued employment. Higher pay and better opportunities in private industry lure the best college graduates and the brightest trainees away from government, particularly in such highly competitive fields as science, engineering, and contracting.... (President's Blue Ribbon Commission on Defense Management 1986:28)

However, the above speaks only in generalities and does not provide the specific areas in which the personnel are deficient. In order to determine whether computer based technology can help, it is necessary to know the specific types of problems that are occurring. A possible answer is

7

found in one set of DoD manuals. This is the area of technical expertise in the specific areas of design, manufacture, and production. The following extracts from these manuals identify the problem:

> Additionally, we must strive for improvement in the understanding and the timing of the disciplines of design, test, and production. Successfully accomplishing the engineering tasks on schedule is the important "key" to reducing the risk of a program. This has a direct and profound impact on the quality of decisions we make on individual programs, and in my judgement, has a more immediate and potentially much greater return on investment in time and effort (and thereby on both cost and performance as well). Most importantly, we can achieve this return on investment with the application of current policy cited in the parent document to this Manual (DoD Directive 4245.7) and using established procedures within the presently defined acquisition process. (U.S. Department of Defense 1985:iii)

> The industrial processes of design, test, and production are poorly understood both by the government, which contracts for them, and industry as a whole, which developed them. That is, some contractors are knowledgeable in, and make good use of certain processes, but no contractor chooses to use them all. As a result, various technical issues in design, test, or production degrade performance and readiness in service, not the management issues. (U.S. Department of the Navy 1986:1-1)

Given that there is a problem with getting good people with the proper training, the next step is to see what types of problems this lack of experience causes. A list of typical problems encountered in program management and DoD acquisition will allow for an analysis to determine possible causes of the problem. If a strong case can be made that the cause is due to lack of training, then a specific area for computer application will have been identified. Below

8

(Table 1) is a typical list of problems cited in Stephanou's text on program management. In addition, the author's analysis, showing a logical link to a lack of experience and or expertise on the part of a PM or his or her staff, is added. The author does not propose that in each case, the failure or cause of the problem is solely due to the reasons developed. Rather these illustrate how a likely cause may be the lack of experience and or expertise. The author realizes that there are always other causes that in any given specific case are more influential than others. However, in order to improve the acquisition process it is necessary to try to eliminate those causes that are solvable and then work from a new level of competence.

Table 1
APPLICATION OF STEPHANOU TO THE INEXPERIENCE PROBLEM

Stephanou's List (Stephanou 1985:14)	Author's Arguments
1. The basis for the project is not sound (inadequate planning).	1a. The staff and PM do not understand operational requirements. 1b. The PM and staff do not understand the acquisition system so they cannot translate the operational requirement into available, workable system.
2. There is a lack of management/company support (including money and other resources).	2a. Due to a lack of understanding on how important support is to the power of the PM and the perceived support the PM receives, the company does not provide sufficient support.

9

Table 1
APPLICATION OF STEPHANOU TO THE INEXPERIENCE PROBLEM
(continued)

3. Tasks are inaccurately defined.	3a. The PM does not know what is required to accomplish the tasks. 3b. The PM does not know what is to be done next and does not plan for the tasks and therefore does not define them well.
4. Management techniques/ systems are misused (or not at all).	4a. Because the PM does not know what is required he or she does not know what to manage. 4b. The PM does not know what the systems are telling him or her.
5. Communications are (faulty information system).	5a. There is a communication problem due to a lack of common understanding between engineers and the PM. 5b. The PM does not understand what information and or what information systems he or she needs.
6. There is too much shifting of personnel owing to changing priorities.	6a. The PM uses crisis management vice a planned management style. 6b. The PM does not know what comes next in the project so that he or she must react rather than control events.
7. There is failure to take into consideration the varying relative importance of performance, cost, and schedule during the project.	7a. The PM does not understand the overall process of acquisition and can not adjust his or her priorities of performance, cost, and schedule. 7b. The PM does not understand that mistimed emphasis on schedule, cost, or performance can cause greater problems in the long

10

Table 1
APPLICATION OF STEPHANOU TO THE INEXPERIENCE PROBLEM
(continued)

	run, by denying proper study of a problem to determine a good solution.
8. The wrong person is chosen as project manager.	8a. Personnel managers have no idea of the requirements for a PM. 8b. No time is allotted for training a PM before assuming the job.
9. The manager falls prey to temptations of expediency.	9a. Due to a lack of understandings of the inter-relationships of system acquisition, the PM can not tell when a decision will impact a later phase. 9b. Due to a lack of understanding of the inter-relationships of system acquisition, the PM succumbs to pressure from superiors to expedite items.
10. Staffing is poor.	10a. The PM does not understand the fields of knowledge required and therefore can not determine how many personnel are required to manage the project. 10b. The PM can not deter-mine the skill levels required for each job and therefore can not utilize personnel effectively nor identify shortfalls in experience.
11. Project termination is not planned.	11a. Since the PM does not understand the acquisition process, he or she can not foresee failure (i.e. termination) coming. 11b. The PM does not have the knowledge of the process of termination.

11

The above analysis indicates that a logical argument can be made for the fact that a lack of experience could be the likely cause of each of the typical problem areas encountered in program management. The presence of this inexperience might be attributable to the fact that either there is not a sufficient number of personnel assigned to a project, or that the personnel assigned lack the required training and or expertise. In the author's experience, both are all too common on most programs. Furthermore, a combination of these causes is usually present at one time or another. Whatever the reason, a lack of knowledge seems to be a main factor that impacts the management of major weapons systems.

However, a further examination is required to determine if the causes for inexperience are solvable together. In the first case, an insufficient number of personnel need to be able to accomplish the tasks they already know, but do it faster, and then be given assistance in mastering a new task. In the second case, there is a sufficient number of personnel assigned, who need assistance in learning their tasks because of their lack of knowledge or experience. The common thread in both of these cases, is that the personnel involved need assistance in learning new tasks. Therefore, it appears that a common solution is feasible.

Another way of expressing this is that there is a need for tools that increase productivity and assist personnel in

gaining experience more quickly. These two items are exactly what computer automation and expert systems can provide. Therefore, it would appear that the DoD acquisition system is a perfect area to examine the use of a Decision Support System/Expert System (DSS/ES). However, it is necessary to first examine further the feasibility of applying DSS/ES systems to the DoD acquisition process based on the technical aspects of the systems being developed, deployed, and supported.

B. DSS/ES DESCRIPTION

In examining the use of computers to assist in problem solving, it is necessary to determine what type of problem(s) are to be solved. The term, type of problem, refers to the nature/structure of the problem and not its subject area. In the previous section, the subject area was selected. It is now important to look at the structure of problems from a more general viewpoint, since it will determine the ability of a computer application to assist a manager. In the past, computers have been useful for solving very structured, repetitive tasks, with a largely numerical basis. It is only recently that computer hardware and software is being developed to deal with unstructured problems.

To utilize this fact, a framework is needed to allow for the classification of problems into a structured or

unstructured category. A good framework for determining or classifying problems was developed by the management information system discipline. Figure 1 shows this framework, developed by Gorry and Scott Morton using business tasks as an example.

	OPERATIONAL CONTROL	MANAGEMENT CONTROL	STRATEGIC PLANNING
STRUCTURED	Accounts receivable	Budget analysis- engineered costs	Tanker fleet mix
	Order entry	Short-term forecasting	Warehouse and factory location
	Inventory control		
SEMI- STRUCTURED	Production scheduling overall budget	Variance analysis- acquisi- tions	Mergers and
	Cash management	Budget preparation	New product planning
UN- STRUCTURED	PERT/COST systems	Sales and production	R&D planning

Figure 1
Information Systems: A Framework
(Gorry and Scott Morton 1971:55)

This framework provides a tool for determining the type of computer application that should be used for a given problem. In general, the development of specific software tools that automate and speed up the execution of everyday tasks and analyses are fairly well under development or

14

already exist. The Defense Systems Management College (DSMC) software packages and commercial packages for such areas as project management, cost, schedule, etc, work well when used by a trained staff. These would correlate to the operational and management control areas for the structured and semi-structured problems.

However, there exists little in the way of computer applications that assist in the solution of strategic planning, unstructured problems. These problems are ones in which the computer needs to simulate the human mind in an attempt to solve the problem. They are ill defined, ill structured, and usually require a large amount of speculation, and imagination just to formulate the real problem. In addition to purely isolated strategic planning, unstructured problems, the author feels that this category should also include problems involving the integration of distinct operational or management control, and structured or semi-structured problem efforts. The reason for this is that integration of a number of fairly simple tasks, that are easily automated, often can not be integrated into a cohesive system due to synergistic, and obscure interrelationships.

Since the management and training of personnel to increase productivity is the problem area selected in this thesis, it would appear that exploration into the development and use of computer applications to assist the

15

new PM could be a step in the right direction. Fortunately, there has been a class of computer applications developed recently that address these types of problems. They are called Decision Support Systems (DSSs) and Expert Systems (ESs). In order to understand this class of applications it is necessary to begin with their definitions.

To begin with, R. H. Sprague defines Decision Support Systems as:

> ...A DSS is a class of information system that draws on transaction processing systems and interacts with the other parts of the overall information system to support the decision making activities of managers and other knowledge workers in the organizations.... (Sprague 1980:12)

DSSs have grown out of earlier Management Information Systems (MISs), in an attempt to develop systems that assist in the solution of all types of problems. DSSs differ from the traditional transaction systems in that they are geared to solving problems that are not deterministic in nature. That is, there is no single solution, or the problem input variables have a range of values and therefore, such that the solution may result in a range of values. The key here is that the DSS seeks to support the decision maker rather than produce a single "correct" solution that only needs to be executed. In this manner, the computer can be used to provide the decision maker with alternatives based on various inputs, the decision would then be left up to the manager based upon his or her evaluation of the

16

alternatives. This evaluation would require an examination of the tradeoffs in both the input variables and the range of solution values.

But this description does not explain how a DSS operates, and without understanding how a system works it is impossible to know how to apply it correctly. Typically, a DSS consists of three components: a dialog, data, and a models subsystem. The user will engage the dialog subsystem and determine what data is present or required. Then, based on what decision he or she is attempting to reach, select an appropriate model and run it based on the existing data and changes or ranges of interest. In this way, the dialog subsystem runs the DSS based on the input of the user, and the data and models are selected and run according to the needs of the user.

A simple example of this is an interest rate problem. If the interest rate changes from 10 to 15 percent, how does that affect a 30 year mortgage payment. The model is the compound interest formula, and the data is 30 years and 10 to 15% in increments. The DSS may have a fixed or flexible percent increment, or it may be that the decision maker wants to first do a course increment (1%) followed by a fine increment (1/4%) to finalize the decision. In this manner the user is shown a range of solutions and can see the impact of changes in the input on the output of the model. This is a simple example, but one can see how this could be

combined with other models to decide say whether to buy or rent a house.

It is the removal of the requirement for a final correct solution that allows the computer to be applied to problems that are ill-structured, or ill-defined. This is because the computer is being asked to do what it does best, compute many repetitive, scenario calculations for interpretation by a decision maker. Yet without a DSS, the decision maker would not always invest the time necessary to investigate the full range of alternatives available and therefore, might miss the most promising alternative. Because DSSs use deterministic models with varying inputs, they are limited in scope or application only by the models contained by the DSS.

At the same time as DSSs were being developed, Artificial Intelligence (AI) was being heavily researched. One of the results of this research has been expert systems. These systems are an attempt to mimic human ability in a specific knowledge area. Don Waterman describes expert systems as follows:

> Expert systems are sophisticated computer programs that manipulate knowledge to solve problems efficiently and effectively in a narrow problem area. Like real human experts, these systems use symbolic logic and heuristics--rules of thumb--to find solutions. And like real experts, they make mistakes but have the capability to learn from their errors. However, this artificial expertise has some advantages over human expertise: It is permanent, consistent, easy to transfer and document, and cheaper. In sum, by linking the power of computers to the richness of human experience, expert systems

enhance the value of expert knowledge by making it readily and widely accessible. (Waterman 1986:xvii)

The above sounds very exciting, as does all new technology, however, it is important to understand how expert systems really work in order to understand what types of problems they can be used to solve. Typically, an ES consists of three components: a knowledge base, an inference engine, and a user interface. The inference engine controls the process and since the problem is known, searches the database for appropriate data. Whenever the inference engine needs data that is not present in the database, the request is sent to the user via the interface subsystem. After receiving the user response, the inference either continues searching or reaches its conclusion.

The main ingredient of an expert system is the knowledge base. Expert systems use knowledge rules to represent expert knowledge gathered from an expert. The format of these rules take on slightly different forms based upon the application, but almost all current representations are based on the "IF... THEN..." statement. This statement allows for the querying of the user for information and allows the computer to conclude some fact based on the rule. By concatenating these rules, it is possible to build systems that guide the user through complex problems and reach a logical conclusion that fits the input data.

Therefore, the user of an expert system will be asked a series of questions that an expert would ask, and based upon the responses would be told the conclusions that an expert would reach based upon the data. This allows for the replication of expert human knowledge. In addition, by observing the steps that an expert would follow, the new user is afforded an opportunity to learn experience at an accelerated rate. It is for this fact, to assist in imparting knowledge that most expert systems offer an explanation feature to allow for the explaining of the reasons for the question and conclusion.

A simple example of this is a diagnostic problem. If a car does not start, what are the steps to determine the cause and can it be fixed? The ES might first ask does the car crank? Based upon the response the system will branch to a different set of questions and or actions, i.e. if yes then check spark, if not then check battery. The question for the spark alternative might be, Do you have engine analysis equipment? Based on the response the system would either say call mechanic (no) or set up and run (yes). In this manner the user is guided through the steps that a mechanic would use to determine the cause of a specific, but complex, problem; a car not starting. And a user with the rudimentary skills or knowledge of cars, i.e. what is a battery, spark, engine analysis equipment, can be shown how to apply that basic knowledge.

20

The example above is just one example of an ES and its method of reaching a conclusion. The method of reaching a conclusion is called the control structure and there are many different types. The control structure to be used is a critical choice since it determines how the expert system will operate and what type of problems the user can expect the system to solve. It is beyond the scope of this thesis to describe all of the different types of control systems, however, Table 2 presents a summary of the various types of control schemes along with their related uses.

Table 2
CONTROL STRATEGIES THAT ARE COMMONLY APPLIED IN VARIOUS APPLICATION AREAS
(Wolfgram, Dear & Galbraith 1987:83)

CONTROL STRATEGY	APPLICATION
Forward Chaining	1. Forecasting, projecting
	2. Predicting
	3. Designing
	4. Planning
Backward Chaining	1. Diagnostic
	2. Monitoring
	3. Controlling
Means-End	1. Synthesizing
	2. Normative Forecasting
Least-Commitment	1. Applications with non-effective pruning rules
	2. Applications with Large, factorable solution space

In studying DSSs and ESs it is difficult to determine where one begins and the other ends. Indeed, there is considerable controversy over this point. Are ESs a subset

of DSSs or are DSSs primitive ESs? A large part of this controversy arises due to the nature of their development. DSSs were developed by MIS personnel to assist the decision makers in their organizations. Therefore, the DSS developers are close to the user. ESs were developed in the laboratory and now are seeking to reach decision makers to prove what they can do. This difference in developmental origin, has given rise to debate over how to classify these computer systems. Turban and Watkins give an excellent synopsis of the opposing views in their paper "Integrating Expert Systems and Decision Support Systems" (Turban and Watkins 1985:138-152). In the author's opinion, a resolution of this conflict is important because it can determine how DSSs and ESs are designed, supported, controlled, and introduced into an organization.

Based upon the definition of both systems it appears to the author that the DSS definition is broader and seeks to address many more types of problems. The methods of DSS are not fundamentally powerful or revolutionary. However, they seek to harness a man to a machine to help interpret more data in different ways. The success of the application is largely driven by the man. However, AI has not reached the state where it can address more than specific problems where expert knowledge exists. Based upon this state of affairs, the author prefers the structure that allows for ESs to be a subset of DSSs. This recognizes that ESs have limitations,

22

yet are an important part in developing systems to assist in decision making. It also realizes that the larger goal is to determine how to assist all types decisions. Therefore, the author views expert systems as a subset of DSSs, used where the intent is to teach or supplement the knowledge of the user. Since this is exactly the type of problem that is being addressed, this thesis will use the term expert system from now on.

C. FEASIBILITY OF USING A DSS/ES

Now that the problem area has been identified and the theoretical background of DSSs/ESs has been established, the next step is to determine the feasibility of using an expert system approach to solving acquisition problems. The purpose of this section is to determine whether the problem area is truly suited for having an ES developed. The author will attempt to follow the discussion of DSS/ESs to show that at each point DSSs/ESs fit the problem area.

As discussed earlier, the information system framework of Gorry and Scott Morton (Gorry and Scott Morton 1971:55) provides an excellent categorization scheme for problems encountered by managers in any field of endeavor. To show how this can be applied to DoD acquisition, Figure 2 is presented as an example of how to apply this framework to systems acquisition. This figure contains representative tasks that have been filled in to show typical acquisition

tasks and their relative structure. This figure is not intended to be exhaustive; but it does illustrate that the information system framework can be applied to DoD acquisition.

	OPERATIONAL CONTROL	MANAGEMENT CONTROL	STRATEGIC PLANNING
STRUCTURED	Data item review	Data requirements Generation of specifications	Acquisition plan
SEMI-STRUCTURED	Cost analysis	Budget planning, scheduling, and design reviews	Contract planning
UNSTRUCTURED	Integration of a data area, i.e. design	Problem identific-ation	Requirement validation

Figure 2
Information Systems Framework
(Gorry and Scott Morton 1971:55):
Applied to DoD Acquisition

The next step is to determine what are the prerequisites for the application of an ES. At first this seems to be a difficult task, but fortunately there exist several checklists that enable one to determine when an ES is appropriate. Both Waterman (Waterman 1986:129) and Wolfgram, Dear and, Galbraith (Wolfgram, Dear & Galbraith 1987:148) provide such lists. Although there is some overlap in these two lists, the author feels that each

24

offers its own advantages. Wolfgram, Dear, and Galbraith's list is general, in purpose, and addresses all of the aspects required for an ES. Waterman has provided three lists, each pertaining to a specific aspect of an ES. Therefore, Waterman's lists allow for the determination of what aspect is not being met, almost an ES itself. Both Waterman's and Wolfgram, Dear, and Galbraith's lists are provided below, along with the author's argument for applying them to this problem.

Table 3
APPLICATION OF WATERMAN'S ES REQUIREMENTS TO ACQUISITION

Waterman's List (All of these are required) (Waterman 1986:129)	Author's Arguments
1. Task does not require common sense.	1a. Acquisition rules are usually specific and not general in nature, or requiring application out side of the specific area.
2. Task requires only cognitive skills.	2a. Program Management is a cognitive vice physical skill.
3. Experts can articulate their methods.	3a. Experts are able to generate manuals, therefore they should be able to articulate their expertise.
4. Genuine experts exist.	4a. In both industry and DoD a limited number of experts are available.
5. Experts agree on solutions.	5a. There is at least general agreement, since DoD manuals, Directives and policy is generated.

25

6. Task is not too difficult.	6a. Acquisition is difficult, however large, difficult problems can be broken up into smaller units, each with its own ES.
7. Task is not poorly understood.	7a. The theory of program management is well developed, it is the application that is lacking.

Table 4
APPLICATION OF WATERMAN'S ES DEVELOPMENT
JUSTIFICATION TO ACQUISITION

Waterman's List (Only one or more of these are required) (Waterman 1986:130)	Author's Arguments
1. Task solution has a high payoff.	1a. Any improvement in acquisition will have a large dollar savings. 1b. Payoff due to better equipment is incalculable.
2. Human expertise being lost.	2a. Government has trouble attracting and keeping trained personnel.
3. Human expertise scarce.	3a. Not enough human expertise exists. 3b. The training of the acquisition work force was mentioned earlier in the Packard Commission Report.
4. Expertise needed in many locations.	4a. The large number of military acquisition, spread over the country requires a large number of experts.
5. Expertise needed in hostile environment.	

Table 5
APPLICATION OF WATERMAN'S ES
CHARACTERISTICS TO ACQUISITION

Waterman's List (All of these are required) (Waterman 1986:132)	Author's Arguments
1. Task requires symbol ismanipulation.	1. Each program different, requiring information to be symbolically represented.
2. Task requires heuristic solutions.	2. Because each program is different, the knowledge will be applied or weighted differently each time, this requires that the base knowledge be applied in a heuristic manner.
3. Task is not too easy.	3. Program Management is complex enough to require years of training and study.
4. Task has practical value.	4. The improvement of management will improve the DoD acquisition system, which in turn will have a practical value to the nation.
5. Task is of manageable size.	5. By breaking the management problem up into smaller units, management is achievable.

Table 6
APPLICATION OF WOLFGRAM, DEAR, AND GALBRAITH'S
ES REQUIREMENTS TO ACQUISITION

Wolfgram, Dear, and Galbraith's List (Wolfgram, Dear & Galbraith 1987:148)	Author's Arguments
1. The problem is well-defined not too large and not too small.	1. The acquisition process has been designed to be modular and hierarchical in nature so that individuals can master parts of it.
2. The domain is reliable, relatively stable, available, and complete.	2. The present acquisition cycle has been developed to be reliable and relatively stable, and well documented.
3. The domain is represent-able, that is, it can be computer knowledge data structure.	3. Structured knowledge is easily represented by computer memory. The anticipated heuristics areenvisioned to be simple "if...then" rules.
4. The data required to be inputted to the expert system for analysis are reliable, available, and complete.	4. There exists a wealth of published knowledge on this topic. This knowledge is in the form of Manuals, Data Item Descriptions (DIDs), and Military Standards (Mil-Stds).
5. The thought process of the expert is not "common sense".	5. There is a lot of "common sense" in the application of the DIDs and Mil-Stds. However, the tailoring of these to each particular system requires the use of expert knowledge.
6. One overall control strategy is capable of solving a majority the domain's problems.	6. Since the goal of the acquisition system is to acquire systems in a well thought out manner, the predictive control strategy should be able to solve most problems.

28

Table 6
APPLICATION OF WOLFGRAM, DEAR, AND GALBRAITH'S
ES REQUIREMENTS TO ACQUISITION
(continued)

7. Users exist.	7. The DoD does not have enough experts yet purchases a large amount of equipment yearly. There exist a large number of users who require assistance in the acquisition of systems.
8. The source of the expertise is recognized as an authority on the subject matter and is readily available.	8. Experienced and senior military acquisition personnel do exist. They may have been since reassigned, but can be reached. Civilian acquisition personnel are easily reached since they do not move around as much.
9. The knowledge is symbolic and not data intensive.	9. One of the major aspects of expert acquisition knowledge is the relationship of the various fields to each other. This is a highly symbolic problem.
10. The application is bottlenecked by existing methods, and only a few good experts exist.	10. The present system is viewed as too complex and cumbersome. This is because few experts exist who understand and are trained in the present system. DoD manpower constraints have made it difficult to develop enough experts. However, some trained experienced personnel do exist.
11. Management commitment is sufficient to support the application selected and to allocate the appropriate amount of time and resources to the development of the system.	11. With the increased scrutiny of Congress and budgetary constraints it has been recognized that better ways to do acquisition are necessary. If a system can be shown to be effective it will be supported.

Table 6
APPLICATION OF WOLFGRAM, DEAR, AND GALBRAITH'S
ES REQUIREMENTS TO ACQUISITION
(continued)

12. If multiple experts exist, then the typical domain problems can be solved with a general consensus among the experts; otherwise, it is not a viable application.

12. There exists one established, published set of guidelines for acquisition. Any disputes will be the result of applying these guidelines to a particular type of system.

13. The organizational culture is sufficiently attuned to accepting and integrating new technologies and innovations.

13. Since the purpose of acquisition is to bring new systems into use, there exists a ready acceptance to new methods.

Based upon the above, it would appear that the use of an expert system holds great promise for helping solve this problem. However, an important factor in the decision to acquire any system is the benefit that can be realized from the use of the system versus the resources utilized in developing the system. Unfortunately, it is difficult to accurately forecast the amount of resources required to develop an expert system. The reason for this is that the resources required is directly related to the design utilized. It is therefore difficult to state categorically what the absolute benefit will be.

However, by surveying the range of resources required to develop different ESs, it will be possible to get an idea of what will be required. ESs in general have three main resource areas; manpower, hardware, and software tools. In

order to get an estimate of the order of magnitude of an expert system development Tables 7 & 8 and Figure 3 show what ESs require in the software and manpower areas. The hardware costs fluctuate so much that a chart would outdated as soon as printed.

These tables and figure illustrate that there is considerable variation in the resources required to develop an ES. It is therefore not easy to pick one of the above approaches arbitrarily, since each approach has consequences that must be considered. Once these design considerations have been made and an initial system design generated, a complete benefit analysis can be conducted that will allow for the easy comparison of cost and benefits.

Table 7
NUMBER OF PEOPLE REQUIRED TO BUILD AN ES
(Wolfgram, Dear & Galbraith 1987:153)

1. One or more senior knowledge engineers.

2. One or more knowledge engineers, either novice or experienced.

3. One or more knowledge paratechnicals (less training than knowledge engineers, but useful in some types of knowledge acquisition, coding, and documentation).

4. Technical management.

5. Project leader (usually a senior knowledge engineer).

6. Programmers, if an AI language is selected, or if the expert system is to be networked with other systems or programs.

7. And, of course the expert(s).

31

Table 8
TYPICAL ES SOFTWARE COSTS
(Wolfgram, Dear & Galbraith 1987:154)

SOFTWARE TYPE	COST
AI language	$6,000-18,000 + hardware + development
Microcomputer tool	$250-10,000 + hardware + development
Mini-mainframe tool	$25,000-40,000 + hardware + development
Prepackaged	$100,000

```
        10,000 | (LISP,PROLOG)
E              |
N              |
G        1,000 | Programming
I              | Environments
N              | (INTERLISP)
E              |    :
E         100  |    :        Early Tools
E              |    :        (OPS5)
R              |    :          :
I              |    :          :        Commercial Tools
N          10  |    :          :        (TIMM,M.1)
G              |    :          :          :      Knowledge
               |    :          :          :      Acquisition
H           1  |    :          :          :      Tools (No Know-
R              |    :          :          :      ledge Engineer)
S              |    :          :          :        :      Full
/         0.5  |    :          :          :        :      Natural
R              |    :          :          :        :      Language
U              |    :          :          :        :      Tools
L              |_____
E            1970    1975    1980    1985    1990    2000
```
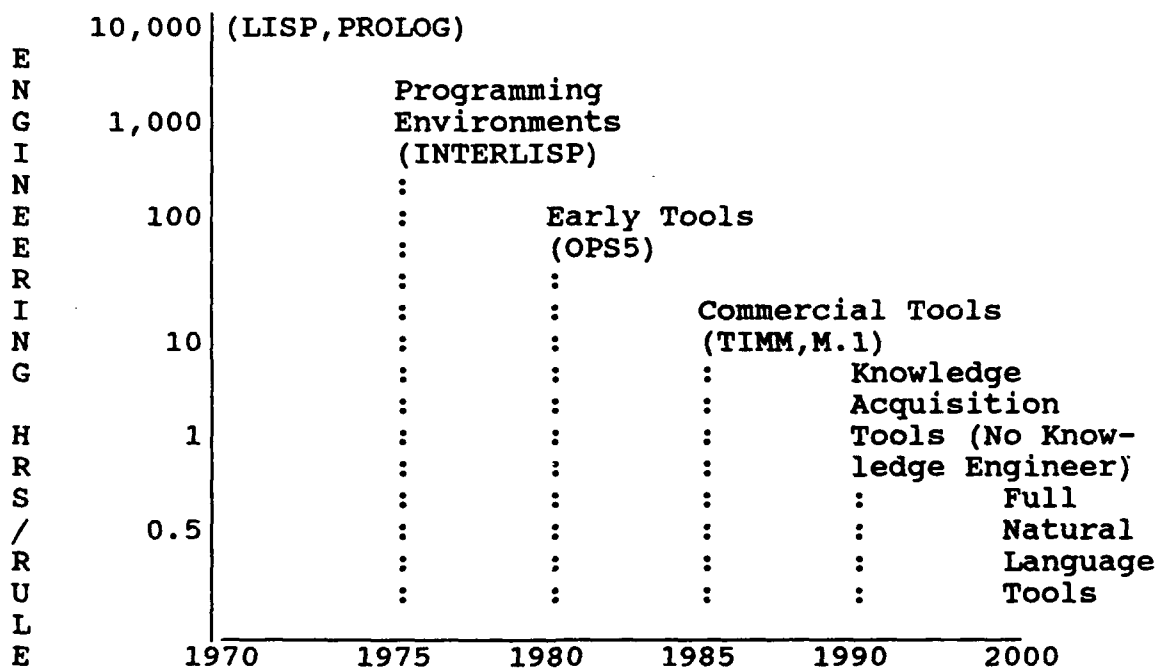
Figure 3
Development Time Hours/Rule
(Adapted from (Wolfgram, Dear & Galbraith 1987:155))

Now that it has been determined that DDSs/ESs fit the
theoretical framework for application to the acquisition
problem, a suitable design architecture and approach should
next be developed. This will further explain the goals of

the system and the use of it. However, it should be cautioned that not all problems will be resolvable at this stage. Indeed it may be that further examination will raise more issues that require more study or are unsolvable. This is because unfortunately, the use, design, and development of DSS/ES systems is an art that attempts to support and emulate the most complex processor known, the human mind.

D. ES SYSTEM ARCHITECTURE

The previous section demonstrated how an ES could be applied to a DoD acquisition problem such as a lack of trained and/or experienced personnel. The next step is to develop a system architecture for an ES to solve this problem. However, there is one more point that must be discussed before developing an architecture, since it directly impacts on the ability of the ES to be developed efficiently and operate effectively. The DoD acquisition system is extremely complex and therefore too large in scope for an ES to handle.

Both Wolfgram, Dear, and Galbraith (Table 6 #1) and Waterman (Tables 3 #6 and 5 #5) state that a problem must be manageable in size for an ES to be developed. Unfortunately, the problem of trying to build an ES for all of DoDs acquisition problems is too large for an ES. This is due to the large number of specialty fields involved with any given acquisition. This would mean that the ES would

33

have to deal with such diverse fields as cost analysis, electronics, spare parts, and contracting. In addition, the ES would have to support the PM and his or her staff. Somehow, the scope of this problem must be pared down before a realistic ES architecture can be developed.

A partial solution to this size problem comes from an analysis of the typical organizational structure of a program office in the context of the information system problem framework. At the top is the Program Manager. Working directly for him, are personnel dealing with the various specialties required to develop the system, such as software engineering, cost analysis, Integrated Logistics Support (ILS), hardware engineering, and systems engineering. Not usually working directly for the PM, but equally important, is the contracting office. Therefore, the typical PM office is organized in a two tier system; the top level consists of the PM, with the second level supporting the technical areas. These two levels correspond almost one-to-one with the information system operational and management control problem structure.

This is an important point because it allows the acquisition problem to first be segregated into two levels. The first level would correspond to the operational control level, and would support the PMs staff. The second level would correspond to the management control level, and would support the PM himself. Therefore, developing an ES with

two levels, the operational and the management control levels, would fit the typical acquisition staff. It also makes the PM support portion of the problem more manageable in size.

However, the staff support portion is still too large to be manageable. Once again, however, the problem is structured so as to provide a solution. Currently, these support fields are considered to be isolated in their applications. In the author's opinion, this is one of the main faults with the implementation of the acquisition system, the lack of a systems approach. However, while not optimal, the present approach, does allow for the partitioning of the support staff level into separate units, with an independent ES for each. This is a first order approach and does not solve the problem completely. But it allows a familiar setting to be retained, thus facilitating acceptance and use, and allows for the problem to be broken up into manageable increments. Lastly, it provides for the increase in the knowledge of the current workers.

An example will illustrate the current manner in which program offices work and the way a "first order" ES will support it. The systems engineer is responsible for the application of Mil-Std 490, the standard governing the development of the system specification. If the Mil-Std is correctly applied, which the system engineering ES will help ensure, it does not interfere with the software engineers

application of Mil-Std 2167, the standard governing software development. The software engineering ES will also help improve the manner in which Mil-Std 2167 is applied.

Therefore, a "first order" ES structure would accept the current practise of supporting each support discipline as separate. This support would be a series of modules such as that of Figure 4. The PM would have a module that assists both in the management of the staff and the progress of the program. This approach will allow for the increased training of existing personnel, and will allow for a better exploration of the interrelationships between the disciplines. The follow-on or "second order" structure would then support these interrelationships between the support disciplines and would resemble that illustrated in Figure 5.

Based on the author's experience, this approach not only breaks the support problem into manageable units, it also makes the problem feasible from a technical viewpoint. This is due to the fact that the synergistic effects of related disciplines are the most difficult to identify. Indeed, it is almost impossible to get the "experts" to agree on what the effects are because each expert is colored by their own background and experience. The ILS expert feels that all synergistic effects are due to the improved support of ILS. Therefore, to attempt to solve both the lack of base knowledge problem and the synergistic effects problem will

be vastly more difficult. In the author's opinion, the DoD has not reached the point where the synergistic effects of acquisition can be determined. Only by first getting enough personnel trained and properly supported, at a minimum level of competence, will it be possible to gain the knowledge base that will allow for the determination of the synergistic effects of these disciplines.

It should be noted that the above structure does not address the strategic planning level of decisions. This is not an oversight, but a realism that in the area of strategic planning for acquisition there are too many political factors that change constantly and will not allow for an easy development of a system to support strategic decisions. In this area it is arguable that there are no experts to provide the information, because no standard method is used to select weapon systems and decide resource allocation. Therefore, this paper will not address the use of DSS/ESs in solving strategic planning issues.

After accepting the structure of Figure 4 as a basis for the development of an acquisition ES, the next step is to determine what further requirements are needed to produce a workable structure. Although each module will have unique specific attributes, it turns out that each of the individual modules will have three main characteristics in common.

The first characteristic is that each module will be a combination of conventional and expert software. There is no need to recreate the large amount of conventional software that has been developed. Furthermore, there are tasks best suited for solution by conventional software. This is not a problem, since current ES technology allows for the interface with some conventional software, and at the least the reading of files of data. Therefore, each module should be considered to be a system of both conventional and ES software tools. The eventual goal should be to integrate these into a single package under the control of an ES. But for now, during the first order development, it is more important, easier, and efficient to build an ES that relies on the manual execution of other software tools for inputs.

The second point is that the first order implementation should rely on existing DoD documentation for its rule base. To this extent, the ESs should be a collection of smaller ESs each based on a particular Mil-Std, Military Specification, or DID. The only attempt at integration of these should be a codification of the existing relationships between these documents, i.e. DIDs to Mil-Stds, or the time phase requirements for them, i.e. C specifications, which specify product requirements, cannot be delivered before B specifications, which specify development requirements. The

rationale for this is again to get the basic knowledge out now to improve the quality of work.

The third point is that the each of the modules will consist of two parts. The first part will deal with the management of the personnel assigned to work in that discipline. For the PM, this module would provide support for managing his or her staff. For a particular discipline, this module would assist in managing the discipline staff, if one exists, or assist the individual expert in managing his or her work. This portion of these modules will consist of time management tools, action item tracking, management aids, etc. The second part will deal with area of expertise. This portion of the module will be a combination of expert and conventional, already developed software. This combination will help the individual PM or staff member determine the technical status of the discipline or overall program.

A final issue in determining a satisfactory structure for the ES architecture is the user environment. There are two primary influences that the user environment imposes on the system. The first of these requirements is accessibility. The DoD acquisition system operates all over the country, with PMs in varying locations. In addition, the PM needs to interact with a number of different organizations, such as service agencies, in-plant representatives, auditors, and contractors. Usually, these

agencies are scattered around the country. Logically, this requires a system that is either portable or else provides nation wide access. The second requirement is to ensure the continued correctness and consistency of the advice given by the ES, to all users. In order for the ES to accomplish this, there must be a standardization of the application of directives to the acquisition process. This requires control over the knowledge base. To allow for each user to modify or develop their own ES would circumvent the limited number of experts in DoD and perpetuate the current system.

In order to meet these two requirements, the ES structure must allow for control of the knowledge base and provide either portability or nation wide access. The easiest and most obvious solution, is the use of laptop computers. In this case the ES would be designed to run on a laptop computer and provide the PM and staff with advice wherever they are. While such a system provides portability, it creates a major problem in configuration control of the knowledge base. This solution runs the risk of allowing the knowledge base to quickly become outdated, with a very difficult problem of issuing changes.

A better solution is to create a central mainframe computer system that contains the knowledge base and allows for the access via a modem. Such a system could allow for the ES to be either run on the mainframe or downloaded to a personnel computer (laptop or desk model). This would allow

40

for easy control of the knowledge base, but would require substantial modem engineering and phone costs. However, the use of the Defense Data Network (DDN) would eliminate the requirement for modem engineering and phone costs. Therefore, the mainframe should be connected to the DDN to allow any user of DDN to access the ES. This architecture solves the dual requirement of access and control of the knowledge base.

```
                            PM
                          MODULE
  ┌──────────┬────────┬──────────┬────────┬──────────┬──────────┐
SOFTWARE     ILS     HARDWARE    COST    SYSTEM     CONTRACTING
ENGINEERING  MODULE  ENGINEERING MODULE  ENGINEERING   MODULE
MODULE               MODULE              MODULE
```
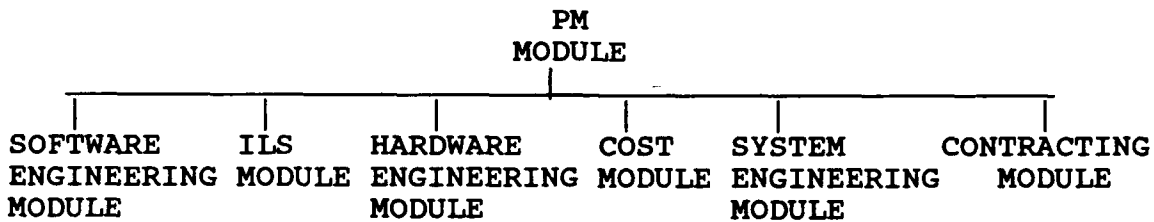
Figure 4
First Order ES Structure

```
                            PM
                          MODULE
  ┌──────────┬────────┬──────────┬────────┬──────────┬──────────┐
SOFTWARE     ILS     HARDWARE    COST    SYSTEM     CONTRACTING
ENGINEERING  MODULE  ENGINEERING MODULE  ENGINEERING   MODULE
MODULE               MODULE              MODULE
    ^          ^         ^          ^        ^            ^
  └──────────┴────────┴──────────┴────────┴──────────┴──────────┘
```
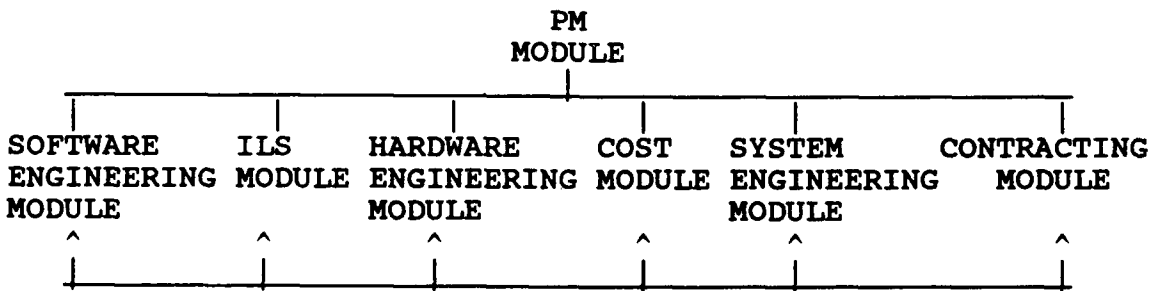
Figure 5
Final ES Structure

III. EXPERT SYSTEM (ES) DEVELOPMENT ISSUES

A. INTRODUCTION

The purpose of this section is to develop more fully the ES system structure in its final form. To this end, this section will explore the issues of what hardware and software should be selected, how it will be set up, used, and maintained. This section is written with the networked dial-up structure in mind, however, this is only a preliminary structure. Therefore, issues are explored with this in mind, but in some instances no conclusion can be reached without further design and prototyping.

This section was originally developed as part of an unpublished paper for a course at the Naval Postgraduate School (Drake and Minnema 1988:4-18). This paper was the joint effort of the author and LT Robert G. Drake, USN. This section was reedited by the author for inclusion in this thesis and was not reviewed by LT Drake.

B. HARDWARE DEVELOPMENT ISSUES

The architecture selected for the ES and any additional user requirements determine the necessary capabilities of the hardware. To recap, the selected ES architecture is one of a central mainframe, containing the program and knowledge base, connected with a remote set of users. Each user will

access the central computer via a modem and download the program and knowledge base. The user will then run the ES on his or her Personal Computer (PC). In order for this structure to be accepted by users, one additional requirement is necessary, speed. Users will balk if the system takes a long time to access, download or execute. However, this brief description is not sufficient to allow for the selection of hardware. Each of these general requirements must be more fully developed in order to allow the generation of a selection criteria for the hardware.

The use of a mainframe is based on two conflicting requirements; speed for the user and centralized control for the Department of Defense (DoD). Both of these requirements are important. Control is required because DoD policies change and these changes must be promulgated and implemented quickly and easily. In addition, a goal of the DoD is to standardize acquisition directives and to ensure compliance across the entire DoD acquisition system. Therefore having many versions or customized versions (tailored by non-experts) of the acquisition ES would defeat this goal. However, the user needs speed so that he or she can expect a near real-time decision aid. This is a very critical factor for the acceptance and use of the system.

The use of a central mainframe allows for both of these requirements to be met. The central mainframe will only be required to perform fetches of the programs and rule for the

43

users. This allows for speed, since the program is downloaded to the user's machine and run there. This also allows for centralized control of the rule base by the DoD acquisition policy makers. In this manner, control of changes to the rule base can be validated and approved prior to implementation. It also ensures that all acquisition managers will have access to the most current data and regulations. "I didn't know about that regulation!", will no longer be an acceptable excuse.

Therefore, the requirements of the central database computer can be summarized as 1) have easy modem access, 2) large online memory capability, 3) suitable security of the database, and 4) to be able to act, for a limited number of users, as a user terminal. The first is to prevent users from having to struggle to get access to the database. The second is to allow for the rapid retrieval and storage of both current and old versions of the rule base. This is a requirement since acquisitions started under one set of guidelines seldom can afford to change to a new of rules set during the acquisition process. The third is to prevent unauthorized access and tampering of the rules. The fourth is to allow for ease of development, testing, and implementation of new versions of the system. Unfortunately, these requirements cannot be quantified until an estimate of the program size is made. However, these

qualitative requirements are sufficient to indicate the types of hardware that could be suitable.

As with the mainframe, the requirements of the user hardware are difficult to predict before a prototype is developed. This comes from the fact that for speed in running complex ESs it is sometimes necessary to utilize symbolic machines, vice standard Von Neumann machines. These machines would represent a significant cost to the development of the system and would limit the use of the system since new users would have to purchase new hardware before using the system. Therefore, unless the speed of conventional personal computers is totally unacceptable, it would be best to utilize them as the user terminals.

Regardless of the speed issue, four user hardware requirements can be determined: 1) high speed modem capability, 2) large online memory or online storage, 3) graphics capability, and 4) hardcopy ability. The first is to gain access to the database. The second is to allow fast storage and retrieval of the downloaded database and allow room for execution. The third is to provide an easy interface for the user (see interface section). The fourth is to provide a permanent record of assistance and plans developed with the use of the system (Wolfgram, Dear & Galbraith 1987:95).

In summation, hardware must be capable of supporting the ES architecture selected. The structure for the ES is a

45

central mainframe with user PCs connecting via modems.
Based upon this architecture, it is possible to determine
four qualitative requirements for both the mainframe and
user terminals. However, these requirements can not be made
quantitative until an estimate of the size of the ES
software is made. However, these qualitative requirements
do allow for hardware planning to begin.

C. SOFTWARE DEVELOPMENT ISSUES

In the development of conventional software, the term
software development deals with every aspect of the software
project. In the development of ESs the term software
development takes on a slightly different meaning. Because
of its importance, the knowledge base is considered
separately. Therefore, software development in ESs is used
to discuss the management of the software vice the actual
contents of the code. Therefore, this discussion of
software development will concern itself with two issues.
These are the choice of a development approach, and the
selection of a set of development tools and or languages.

Once the hardware is preliminarily determined, it is
necessary to consider the software approach to be used
during the development. According to Pressman (Pressman
1987:19-27) there are two main development paradigms for
software. One is the classic and the other is the prototype
or evolutionary approach. The classic approach is best

suited for problems where the requirements can be determined completely apriori (Alavi and Napier 1984:65). The evolutionary approach is best suited for those problems where the end goal is known but the methodology is not known or there is more than one manner in which to achieve the end goal.

In order to choose between these two approaches it is necessary to consider certain software development policies of the DoD acquisition process. Unfortunately, DoD acquisition normally requires an classic approach to developing software. A strict interpretation of this approach has been shown to be the least satisfactory method of developing expert and decision support systems (Hogue and Watson 1984:76; Waterman 1985:135). However, DoD software regulations do not prohibit the use of prototypes. They only require that the use of prototypes be planned for and that the final system be fully tested prior to deployment. It is therefore planned to use a hybrid of the two approaches that will allow for the efficient development of the ES, and yet deliver structured, and maintainable software.

To implement the hybrid approach for modules, it is proposed that the prototype approach be used for initial module development and testing. Upon completion of the prototype, a shift to the classical approach would occur. This will allow for the exploration of different types of

47

development tools, languages, methods of data representation, and interfaces. Most of these aspects of the system can not be specified prior to preliminary data acquisition and interviews with the experts. There would be no limit to the number of prototypes other than cost, schedule, and the skill of the module development team. The culmination of the prototype stage will be the completion of an informal performance test devised by the module team. Upon completion of the initial module testing, development would shift to the classic approach with its detailed design specifications. The module would then be developed in the approved final language, using specified development tools and subjected to formal acceptance testing.

To extend this approach to the entire system, it is proposed that a two stage approach be used. In the first stage, all modules will be developed separately and concurrently by individual development teams, including the PM module. This will allow for the development of a minimum capability system in the shortest amount of time. In the second stage, the modules will be reworked to incorporate any additional knowledge discovered during the first stage. During the second stage, the PM module will be the one requiring the most modification. The rationale for this approach is to allow for the discovery of all potential interrelationships between the modules before attempting to develop the final versions. In the author's opinion, it is

48

highly likely that the development of the first stage modules will demonstrate or discover new key aspects of the management of DoD acquisitions. In any event, it will allow the developers to become more familiar with the problem before they start trying to integrate the functional areas.

The other aspect of software development to be considered is the type of development tools and languages to be used. In earlier development of ESs, specialized languages were written that were more suited to the representation of knowledge and execution of expert rules. Although these languages are extremely powerful and quick in execution, they usually require an experienced programmer and require more development time for the ES. Recently, there have been a large number of ES tools developed to shorten the development time and to allow more novice programmers to develop expert software.

These recent development tools can be classified into three categories: expert languages, expert shells, and prepackaged commercial applications. Expert languages are updated versions of the original languages. Using them means that all tools, interfaces, and parts of the ES will have to be developed from scratch. Expert shells are an attempt to establish a basic ES that will support any knowledge base installed. This significantly reduces the development time for the system, but usually is restricted to one type of control mechanism. The prepackaged

49

applications range from entirely developed ESs, to development tools, such as those used to extract the knowledge from the experts. Depending upon the application, it may be possible to purchase already developed systems or to buy the shell and an interviewer package that will generate the knowledge base.

The choice of one of the above tools is dependent upon the ES characteristics. Unfortunately, at this point in the planning it is impossible to determine the control mechanism or the complexity of the knowledge base. Therefore, the only arguments that can be made are for speedy development, reduced development cost, and ease of maintenance. Based upon these requirements, it is proposed to use ES shells and if necessary an expert interviewing system. ES shells will support the rapid development of the first stage modules, and if necessary can be replaced or augmented during the second stage of development.

In summation, ES development requires a choice of the development approach to be used. From the two major schools of development thought, a hybrid approach is developed. This approach will allow for a rapid development by utilizing prototyping combined with informal testing. Upon completion of the prototyping stage a formal development stage will be started. In order to support this approach the use of ES shells will be used to support the development of the ES.

D. KNOWLEDGE BASE DEVELOPMENT ISSUES

The most important part of an ES is the knowledge base (Goul and Tongue 1987:450). Therefore, particular care must be paid to its development. In considering the knowledge base of an ES it is necessary to discuss four topics: the types and structure of available knowledge, the sources of the knowledge, how the knowledge is to be extracted, and the control mechanism.

For the DoD acquisition problem, there are two types of knowledge: general acquisition knowledge and specific application knowledge. The first type deals with general methodology knowledge that explains how to acquire any system. The general acquisition type of knowledge is represented by the regulations and documents pertaining to all DoD acquisitions. Examples of this are the software development standards, Data Item Descriptions (DIDs)., systems engineering manuals, and federal acquisition regulations. The second type deals with the specific application of acquisition knowledge to a specific program or type of program. That means that the application of acquisition knowledge to the procurement of electronic equipment is different from the application to the procurement of ammunition. The specific acquisition type of knowledge is represented by Military Handbooks, Manuals, and experts. It is the specific application knowledge type that contains the most expertise knowledge.

The use of these knowledge types depends upon the source of the knowledge and the development stage of the ES. The general knowledge is readily available in the published DoD directives, standards, specifications, etc. The specific knowledge is spread between published manuals, such as military handbooks, and human experts. During the first stage of ES development, the general knowledge will be used to provide a minimum capability and to raise the level of expertise in DoD acquisition personnel. During the second stage, the specific knowledge, along with the knowledge gained during the first stage, will be incorporated into the ES. With this approach, a general knowledge base can be developed quickly, allowing the specific knowledge base to be built on a solid, working, foundation.

The method of extracting the knowledge depends on its source. The extraction of knowledge from the general knowledge category will be done by the knowledge engineers researching their particular functional area. Because the specific knowledge category consists of both manuals and humans, a combination approach is required. Research by the knowledge engineers, to determine appropriate published material, combined with an initial survey of experts, to determine other relations, will be utilized. Personnel presently in acquisition billets will be the initial survees.

Some further description of the survey process is necessary in order to provide the reader a full understanding of its purpose. The purpose of this survey is to get an idea of the scope of material and types of sources that the experts feel are important. One portion of the survey will also include a request to list other "experts". Upon completion of the survey, use of an automated expert knowledge tool will be used to extract a deeper level of expertise. The data from the survey will be use to set up the interview software. Lastly human interviews will be used as a final step in extracting difficult or contradictory knowledge from the experts. Since acquisition experts deal with documentation and people it is envisioned that the interview method will be most satisfactory.

It is possible for the above approach to be misinterpreted as to the content of the knowledge base. The purpose of the ES is to raise the knowledge level of DoD acquisition personnel. This should not mean the automation of all of the acquisition standards. This would create a large, inefficient, and overwhelming ES. The approach should be for ES to describe what manuals are important and why. In this way the knowledge engineers can develop a system that contains the minimum factual data with references to the remaining published information. This will prevent the system from being cluttered with pure factual data that is already available. However, if certain

standards are deemed prerequisites by the knowledge engineers, they will be included in the system.

The selection of a control mechanism is dependent upon how the user wants to use the knowledge. The same problem and knowledge base can be used in various manners, each of which require a different control structure. For DoD acquisition, there are two primary approaches used in the management of programs. The first is to assist programs already in progress. This entails the use of the ES in a diagnostic manner, requiring a backward chaining control mechanism. The second is to assist in the planning of a program. This is a forecasting or "what if" manner, requiring a forward chaining control mechanism. These mechanisms can be used in the same ES by prompting the user to state what the session is for, planning or troubleshooting. Therefore, the ES should, as much as possible, incorporate both of the control mechanisms.

In summation, the development of the ES knowledge base requires determination of the types of knowledge, the sources of the knowledge, the extraction of the knowledge, and the control mechanism. In DoD acquisitions two types of knowledge exist, general and specific. The sources of this knowledge are found in published documents and human experts. The methods of extraction of this knowledge will be research and surveys of experts. Finally, the control

mechanism will support both the planning and the troubleshooting of DoD acquisition programs.

E. NETWORK DEVELOPMENT ISSUES

In order for the DoD to use a common ES, the use of a network was decided to be the best solution. In particular, the Defense Data Network (DDN) was cited as an existing network for potential use. The purpose of this section is to discuss the ES requirements of a network, and document the advantages of using DDN. The DoD acquisition ES imposes two requirements of the network: allow access to the ES and support the ES interface. The advantages of using DDN will be seen as a substantial benefit to the ES.

Access can be characterized in terms of three things: complexity of connection, difficulty of use, and cost. Complexity of the connection means the hardware and software required to allow use of the network. Some networks require special lines, along with expensive interface equipment to allow communication. Difficulty of use deals with the training required to allow the user to access the network. This is a combination of the hardware and software and reflects the simplicity and reliability of both. The cost is a function of the hardware, software, and operating expenses. That means that if there is a connection or usage cost for the network (i.e., phone call charge, central processing unit time charge, etc) it must be considered.

The use of DDN will allow for maximum access to the ES system. DDN is designed to allow users with standard phone modems to access the network. These modems are fairly cheap, use a standard interface, have high speed (1200 baud), and fit in most PCs. These modems are easy to use and many users already are already experienced in their use. Furthermore, the DDN is structured with local access points nationwide, eliminating expensive toll calls to a central location. Therefore, the use of DDN offers an optimum tradeoff in the three areas characterizing access.

Support of the ES interface can be characterized by two items: support both text and graphics, and allow the transmission of program code. The requirement of text and graphics is due to the nature of the knowledge base. Presently the DoD uses text and graphics to explain relationships and knowledge about acquisition programs. Therefore, the ES must provide this interface in order to be accepted. A textual interface is standard to any network, however a graphics capability is not. However, since the execution of the ES is envisioned to be on the user terminal, the network need only support the transmission of the graphics information in a form usable by the user terminal. This may require conversion from one terminal form to another. The requirement for the transmission of program code comes from the decision to utilize a central mainframe. Current versions of the ES along with data will

be required to be sent to the user for execution on the user terminal. Therefore, a fairly rapid capacity to transmit programs is required.

The use of DDN will provide the interface support required by the ES. The DDN already supports textual and graphical interface. The graphical interface requires knowledge about the terminal in use, but once the terminal is identified, DDN performs all required conversions. Furthermore, DDN was also designed for the high speed transmission of files. These files can contain data, or programs, and are transmitted unaltered. This means that programs can be transmitted and upon receipt, will be ready to run on the user's machine. DDN supports several different protocols for the downloading of files.

Several further advantages come from the use of DDN. Networking can play an important role in the development of the ES. By allowing the ES to come to the experts in their own familiar work environment, it will save the experts time in travel and promote a more cooperative atmosphere. By allowing easy interface between developers and experts, cooperation during the development, and testing of the ES will be enhanced. Since most large Government contractors and installations already have, or can get, access to the DDN, the cost of this solution would be minimal.

In summation, the DoD acquisition ES imposes two requirements on the network. These requirements are to

57

provide access and support the ES interface. Access is a function of connection requirements, ease of use, and cost. The ES interface requires support of text and graphics, and the transmission of programs. DDN is capable of meeting these requirements, and offers several other advantages.

F. INTERFACE DEVELOPMENT ISSUES

One of the more important development issues is the type and quality of the ES interface. The overall effectiveness of the system may be determined by the frequency of its use and the accurate interpretation of the information displayed. "A well designed dialog component does not guarantee the success of a DSS, but it is a necessary ingredient." (Sprague and Carlson 1982:217). However, the judgement of an interface is very subjective to the particular user or class of user. Therefore, the development of the interface must be on a sound basis and be responsive to the requirements of the user. In order to ensure this, the development of the ES interface will deal with the three parts of an interface, and the style of the interface.

Physically, the user interface consists of three parts, the Action Language, the Presentation Language, and the User Knowledge Base (Bennett 1977:3-11). The action language deals with how the user can control the system. That is does the interface allow the user to type on a keyboard, or

use a mouse, or speak to control the actions of the system. The presentation language deals with how the system presents information to the user. That is does the interface present information to the user on a screen, via a printout, or an audio output. The knowledge base deals with what the user must know in order to in order to use the system. This does not refer to the users knowledge of the interface, but rather the knowledge the user needs to solve the problem. An example of this would be that the system expects a user to be conversant in the problem field and therefore, answers are not explained in lay terms.

The acquisition ES interface will fit this same structure. The action language will be either a keyboard or mouse depending upon the user terminal. These two are selected due to their already wide application and relative inexpense. The presentation language will consist of screen displays consisting of text and graphics, with a printer output option to provide a hardcopy record of the session. These mediums are selected due to their use in the management of acquisition systems. The knowledge base will be kept to a minimum. This is due to the fact that a main goal of this system is to educate and train acquisition personnel. It therefore does no good to require a user to already be knowledgeable about acquisition in order to use the ES. The selection of these physical characteristics

should provide a familiar interface to users, which in turn will make the acceptance of the system more likely.

Another dimension of the interface, that impacts all three parts of the interface, is the concept of "dialogue style". The style determines the manner in which the three physical parts of the interface will be used. Therefore, the style is important since certain styles have limitations that make them suitable only to certain problem structures. Sprague and Carlson point out that there are many types of styles and many combinations of them (Sprague and Carlson 82:199). Each style or combination of styles must be evaluated for potential tradeoffs before being selected for a particular application.

However, Sprague and Carlson do cite four examples of styles that, in the author's opinion, cover the majority of present day ESs. These four styles are the questions/answer, command languages, menus, and input form/output form. The question and answer style is simply that the system or user poses a question and the answer is then provided. The command language style requires the user to enter specific commands to control the system, an example of this is PC Disk Operating System (DOS). The menu style allows the user to select a command from a list via the use of a simple input medium, i.e., number, mouse, letter. The input form/output form language style requires the user to enter information in a "fill in the blanks" manner. An

example of this is spreadsheet calculations, the information is entered in the blank or cell located in the form.

The style of acquisition ES will be a combination of two of the above styles, the menu and question and answer styles. The menu style will be utilized to control the system. The menu style reduces the amount of training required for a new user and provides for most visible means of system control. Once control is passed to an expert session the question and answer style will be used. This style is the natural style of consulting with experts. The expert must have specific types of information, known only to the expert, and the user provides it. It is therefore only logical to use the same approach when dealing with a system that is trying to replicate an expert. This combination of styles will allow for an easy to use control system and an effective and familiar consulting style.

In summation, the interface of the ES can be a very important aspect of the use and acceptance of the system. In the discussion of the interface, a three part structure is utilized. The envisioned structure of the acquisition ES discussed in these terms. A further dimension of the interface, the style is also discussed. Using this discussion, the control and consulting style of the ES is determined. The result is an interface that will provide the user with a familiar interface that will assist the ES in being utilized and accepted.

G. VALIDATION

The validation of ESs poses several unique problems. Since ESs attempt to duplicate human problem solving techniques, they are difficult to test in a deterministic manner. Therefore, no series of tests will allow for the determination of whether an ES works correctly or not. Simply put, ESs deal with problems that have no right or wrong answer. Therefore, any evaluation of the system will require the use of experts to determine the correctness of the system (O'Leary 1986:470). Yet lack of a validated ES can lead to a lack of confidence in the system or worse, a system that makes mistakes.

In order to develop a validation scheme that will prevent this, it is necessary for the validation process to support, not hinder the development process. Therefore, the validation scheme must be technically sound, yet support the development approach selected. For the acquisition ES, a further requirement is that the validation scheme support the centralized control of the knowledge base. A validation scheme that accomplishes these things will allow for the determination of the quality of the ES.

There are two approaches used to validate ES software. These are an informal and formal validation. Informal validations, usually do not have a firm set of evaluation criterion, but are used to determine if the design approach is headed in the right direction. An example of this would

62

be review of the rule base with the expert to ensure that the order of execution is correct. Formal validations are structured with a predefined set of evaluation criterion and usually are invoked at the conclusion of a milestone in the development process. An example of a formal validation is the acceptance of a display module for incorporation into the ES. The display module will have a requirement to accept information in a defined format and display that information in a specified user format (Wolfgram, Dear & Galbraith 1987:157).

Even with these validations, it is difficult to determine the pass or fail criterion of the ES. Seldom will all of the experts agree on the application of their expertise in all of the test scenarios. One approach to overcome this, is to use a certain percentage of the experts agreeing that the system operates appropriately as the pass or fail criterion. Presently, a 90-95% level of consensus is discussed in the literature. However, an important measure of effectiveness for an ES is the amount of time that it saves the users. Therefore, any pass or fail criteria must try to measure, or at least take into account, the increases or decreases in training time, work time, or performance.

The above two approaches must also be combined with the development approach and goals of the ES. The development approach has been defined as one of a concurrent iterative

63

development of modules. Furthermore, the goal of the system is to provide education, and assistance to acquisition personnel leading to a standard application of DoD acquisition directives. Therefore, it is planned to have a series of informal validations during the development of modules and a formal validation of each module upon delivery.

The informal validations held during the development of modules will utilize the experts who provided the knowledge. However, the last informal validation will utilize typical users in a series of case scenarios. The use of newly graduated students from the Defense System Management College (DSMC) courses is one possible source. The use of these students offers an excellent opportunity to utilize unbiased, motivated, potential users, who have a rudimentary level of acquisition knowledge. The feedback received will provide the final test of the modules ability to be used, and assist new PMs.

For the formal validation procedure, it is proposed to utilize the the DoD acquisition policy makers. The DoD acquisition policy makers will be used as reviewers of the case scenarios results to determine if the ES accurately implements the present DoD acquisition policy. This will minimize the drain on the policy makers time and yet ensure that the system does not guide acquisition personnel into violating DoD policy. Furthermore, the use of the policy

makers as the final reviewers will ensure their support of the ES and will send an important message to acquisition personnel that the system is approved for use.

In summation, the validation of ESs is a difficult yet important task that usually requires more validation steps than conventional software. Furthermore, it is difficult to determine the pass or fail criterion for the system. A consensus percentage of experts is one method that can be used. For the acquisition ES, the use of DSMC students along with DoD acquisition policy makers will provide a suitable set of experts that will validate the ES.

H. MAINTENANCE AND SUPPORT

ESs are adaptive and iterative in their development and they are never static (Wolfgram, Dear & Galbraith 1987:161). In addition, DoD acquisition policies are constantly changing and therefore, force the ES to be modified in order to remain current. Because of this, maintenance of ES will be required and probably will require a substantial effort. Therefore, the maintenance of any ES software should also be considered during the design and development stages. The lack of this planning will result in a system that is only usable until a change is required and then an entirely new system will have to be developed. On the other hand a system built considering a well thought out maintenance concept, will be easy to improve and keep current.

There are two main issues to consider in planning the maintenance of an ES. The first issue is the standard software maintenance problem of the choice of development tools, selected programming language, and the required skill of the maintainer(s). The second issue is control of the expert knowledge base. Put another way, who are the experts that decide the system is in error and requires fixing? This is a problem peculiar to ESs and is vital if the ES is to support the DoD acquisition problem in a uniform, homogeneous manner.

The acquisition ES has taken the first issue into consideration as much as is possible at this stage. The previous consideration of the various development tools selected the ES shell as the most productive tool. These shells are readily available from commercial sources. The use of a commercial ES shells should reduce the required number of programmers for maintenance. The use of a shell will also reduce the knowledge requirements of the programmers since they will be utilizing a standard development tool, and not having to design a new one for support. Therefore, the development strategy for this ES satisfies the support requirements of software maintenance.

The development of an approach to satisfy the second issue is more difficult. This is because of the additional maintenance requirements of an ES. Both conventional and expert software maintenance requires an activity to perform

the standard functions of troubleshooting, research, coding, debugging, and configuration control. However, ESs also require access to a group of experts in order to allow for the validation of any necessary changes. Since these experts are in short supply (a requirement for the successful development of an ES), it is impossible to capture a group of them and assign them to the software support activity. Therefore, any maintenance plan must take this into account and attempt to minimize the impacts of having experts not readily available.

In order to accomplish this, it is proposed to utilize the following approach. The software support facility will perform all of the standard maintenance functions. Since the DoD has created the DSMC to provide a reference center for acquisition, it would appear obvious for them to be the central focal point for support. The DSMC has a software development group already in existence, working on the procurement of software to assist program management. If this approach were followed, at one location both the software developers and maintainers and experts would be collocated. In the author's opinion, this would be an unusually logical arrangement that is seldom followed. This organization would be able to do the necessary analysis of problems, development of fixes, and testing of these fixes.

However, changes to the system should be approved at the DoD acquisition policy maker level prior to release.

Obviously, these policy makers will not be doing the coding or testing of the changes, but approval of any changes should require a sign off at this level, since they are responsible for the implementation of the various regulations and policies. This is even more critical for this system since one primary goal of the system is to tutor and train the new acquisition worker. The policy maker should therefore ensure that the training tool is kept accurate and reliable.

In summation, the maintenance of the acquisition ES will almost certainly be a continuous and substantial effort. It is therefore important to minimize this effort by planning for maintenance during the development of the ES. The ES development approach selected provides for the reduction of the maintenance effort through the selection of tools require a minimum number of personnel. The maintenance of the knowledge base is more difficult and requires access to a group of experts to validate any changes to the ES. The use of the DSMC software research center, combined with review by DoD acquisition policy makers should provide a satisfactory approach to ensuring the maintenance of the ES.

IV. EXPERT SYSTEM (ES) PROTOTYPE

A. INTRODUCTION

This section will describe the various issues involved in the development of the prototype. It will attempt to parallel the structure of the previous section to show how some of the issues raised were addressed. The purpose of this prototype is twofold. The first is to prove the concept of applying ESs to acquisition, by providing a working system. The second is to demonstrate that existing Department of Defense (DoD) manuals can provide an useful source of knowledge with out a large investment of resources in developing the ES.

B. HARDWARE

Some people feel that the selection of hardware can be isolated from all other considerations. While this can be done it usually leads to increased development of tools that do not exist for the selected hardware. Therefore, the selection of hardware should be closely linked to the software required to accomplish the task. This rule cannot be forgotten if an efficient development environment is to be established. Hardware with out software is useless and vice versa, worse great hardware with bad software is worse than a system consisting of average performance.

69

Based on this, the selection of hardware for the prototype was driven by three considerations. First was the desire to select a hardware that was available to potential users until the mainframe system is set up. The second was the availability of software to run on the selected hardware. The third was the ease of use and access during the development of the prototype.

The first consideration led to the selection of a Personal Computer (PC) based hardware suite. This is due to the fact that almost all program offices have or have access to a PC system. Furthermore, the selection was made to use an IBM compatible system since the Government has selected that as its office standard. A last, though not inconsequential consideration was that the author owns an IBM and is familiar with its architecture and operating system.

As stated earlier, the second and third considerations are closely interrelated. In order to find a suitable hardware suite, it was necessary to determine the hardware requirements of existing expert software. It would do no good to select a hardware suite that was too exotic to assemble. This led to a survey of existing commercial ES shells. Several published references were utilized and offered excellent comparison tables of existing software tools (Waterman 1985:339-365; Wolfgram, Dear & Galbraith 1987:131; Defense Systems Management College 1986:2-2). The

70

result of this survey was that there exist a number of ES shells that are all capable of running on an IBM PC, and that provide a suitable development environment. The most exotic requirement of most was that of a hard disk for large rule bases.

C. SOFTWARE

Based upon the above selection of hardware, a final selection of software was made. The ES shell selected was the M.1 system by Teknowledge. The criterion for this decision was based upon purely pragmatic reasons. The final selection of the system was made strictly due to the fact that M.1 was available at the Naval Postgraduate School (NPS). In addition, there existed sample programs developed by NPS students. This greatly decreased the learning time required for the author to develop a working control structure.

To say this decision was pragmatic does not mean that M.1 is not a suitable choice. The M.1 system, is a robust ES shell, by any comparison to others on the market. M.1 allows for a rule base of virtually unlimited size, due to the remove and load functions. It allows for the inclusion of graphics, external routines written in the C programming language, and allows for external calls to data files via the operating system. In fact, the only real criticism of M.1 is that it does not generate executable code. The

system is interpreted and therefore requires the user to own M.1, however, this is offset by the fact that when running M.1 rules can be added and saved. One last point in M.1s favor is that M.1 was derived from a mainframe ES S.1, also by Teknowledge. This means that the coding on a PC should be very transportable to the mainframe version. If this proves out it would give a very strong argument to examining the use of S.1 as the mainframe ES.

D. KNOWLEDGE BASE

In considering the knowledge base of the prototype, the scope of the work involved became the paramount issue. The restriction of one person attempting to develop the ES prototype, quickly became apparent. This appeared to be fatal restriction, since the purpose of prototype is to quickly develop a partially working system. Therefore, the first decision was to concentrate on the product portion of the system. This is the portion that involves the use of ESs, and development of this portion is needed to prove that ESs can be utilized in DoD acquisition.

Yet a partially completed ES is not feasible, since expertise is not partial. Therefore, in order to develop a prototype that is usable, the author decided to concentrate on a single module and attempt to complete the product portion of it. This will allow for one specific functional area to be supported. However, even one module posed a

72

significant amount of work. Which module to select? Would choosing the software module be better for a prototype than choosing the hardware, or the costing modules? Even with the selection of only one module, the amount of work involved in developing the expert knowledge base and validating it is substantial.

The solution for which module to develop came by thinking about who in the program organization will bring in new technology. More important, who will provide support for the continued development of the entire system? Based on these questions, it was decided to develop the product portion of the program managers module. The reasoning behind this is that the Program Manager (PM) is ultimately responsible for the program and anything that can help determine the state of his or her program will be more readily accepted. Also, if the PM does not trust or accept this technology, then use by his or her staff will probably be limited. This logic is summed up in the line: impress the boss first and the rest will follow.

The problem of the knowledge base still exists. This is the real work in any ES. There has to be agreement on who are the experts, then the knowledge must be extracted from them, put into a working ES, and finally validated against the experts to ensure the knowledge was not corrupted somewhere along the line. This sequence of events is what has led to long development times of large ESs. Faced with

this, the development of a knowledge base to assist the PM in determining the status of the program seemed almost too ambitious.

However, a solution was found that eliminated the need for determining experts, culling the data, and validating the ES. The approach used was to take approved DoD manuals that described typical problems encountered in acquisition programs. Even more fortunate, these manuals also provided detailed reasons why, and symptoms of the problems. The fact that these manuals are approved by DoD means that they can not be ruled inaccurate since the "experts" approved them. Furthermore, since they describe typical problems and not methodology, they are applicable to all programs. The manuals selected were the DoD 4245.7-M **TRANSITION FROM DEVELOPMENT TO PRODUCTION** and the Department of the Navy (DoN) NAVSO P-6071 **BEST PRACTICES**.

An added benefit of the selection of these manuals is the manner in which they are structured. These manuals were broken up into the functional areas involved during the transition development to production process. These functional areas are: funding, design, test, production, transition, facilities, logistics, and management. Each of these areas was itself broken up into specific subareas or topics. For example facilities consists of four topics: modernization, factory improvements, and productivity center. For each of these topics, an explanation of the

74

topic is provided, and four of the most common traps associated with that topic were identified. Each of these traps is discussed by providing the present practice, symptoms, corrective action, and benefits of the corrective action.

The structure of these manuals provided for a fairly easy ES development. The similarity of structure allows the expert module for each topic to be structured the same. The explanation for each topic can be inserted without modification. The listing of the traps allows them to be asked as questions, answerable by yes or no responses. Appendix A illustrates this by containing a commented sample of the main control structure and one module (transition). Appendix B provides a user manual for installation and operation.

Even with the selection of these manuals, there were a number of difficulties encountered in deciding how to develop the knowledge base. The resulting method was often the selection of the method that would ease the development. Unfortunately, it is not possible to determine if these difficulties are critical or not until the prototype is used. It should be remembered that none of these difficulties are irreversible, and that the purpose of a prototype is to quickly determine what works best.

One of the difficulties is the use of a standard structure. This may allow for the user to "game" the ES.

75

This was considered, but for the prototype, the goal is education, not correction, therefore gaming should not be that prevalent. If a similar structure is found to be undesirable, each module can be restructured. This will complicate the development, but will be transparent to the user.

Before discussing any further difficulties,it is necessary to discuss the term trap as used in the DoD manuals. The DoN Best Practices manual provides the following definition:

> ...these approaches, standard ways of doing business in today's defense systems acquisition environment, as "traps" since they represent potential danger to program success. Although traps may not appear to be inherently dangerous, they become problems when they are sprung. There are indicators, or "alarms," both subtle and obvious, which alert the project manager to the fact that he is caught. On the other hand, the dangers of a trap can be avoided if he knows how to "escape." The project will immediately relate to the traps discussed in this manual because with few exceptions he will find them in his project. (U.S. Department of the Navy 1986:1-5)

What this definition says is that certain practices can appear to be correct but in reality are a serious flaw when used incorrectly. An example of this will make it clearer. In the Transition Plan template, trap #1 states "Transition plan is reviewed and approved by government at Milestone III" (U.S. Department of the Navy 1986:7-2). This appears not to be a trap, but a very good idea, for two reasons. First, the Government required that a transition plan be developed and second, the Government is reviewing the

76

transition plan at the same time it is making the decision for production. However, this is exactly the manner in which this trap is sprung. The correct use of a transition plan is to develop it early during the Full-Scale Engineering Development (FSED) phase and to require its use during the FSED phase.

These manuals provide the four most common traps that are prevalent in each of the functional area templates. For each trap in a template, a list of the escapes, along with alarms are listed. If a template is used in the manner of the escapes, it is not a trap. Conversely, if the alarms are observed, then the project has a greater risk of problems. In this manner, the manuals attempt to warn the PM of the risks associated with even the "standard" manner of acquisition. Only by understanding why something is important, can the PM ensure that it is correctly employed.

This introduces the next difficulty. During the research of the prototype, it was decided to use the trap itself as the question vice the symptoms for the trap. This approach was taken for two reasons. First, it allows the structure to be the same, thus speeding development. Second, it stresses the traps themselves. By asking the trap as a question, it is hoped to stress that this trap does occur. Whereas the same symptom can mean two or more problems. Since each trap has a varying numbers of

symptoms, this will mandate that each module be structured uniquely.

The last difficulty is the verbatim use of the manual descriptions of each topic. This could be construed as providing a biased viewpoint and therefore limit the learning ability of the user. Industry and DoD do have different goals and viewpoints on development. By not providing a "balanced" view, the user may be lead to believe that the DoD view is the only method. This is a valid point on the blanket use of DoD manuals. However, the DoN manual was developed by a joint team of contractors and DoD personnel and so this problem should be minimized.

E. INTERFACE

The choice of interface was determined by the lack of time and experience in the use of M.1. Therefore, the standard M.1 interface panels consisting of questions and answers was utilized. This is not to imply that M.1 does not allow for easy modification of its standard interface. M.1 is very flexible in this regard and as earlier mentioned allows the use of graphics. There simply was not time to learn the control aspects of this prototype and develop a new interface.

Therefore, the format of the two manuals was used. Parts of the manuals were used verbatim as the explanation of the topic and the traps themselves were utilized as the

questions. However, one additional feature was added to the manual format. This was the addition of an explanation panel for each question. The reason for this is that many of the traps, when phrased as questions, assume a level of knowledge that may not be present in all users.

F. VALIDATION

As stated previously, the validation of an ES is crucial to its acceptance and success. No one will use an ES that makes mistakes. However, this is also the most difficult part of the development of an ES. For the prototype, it was decided to utilize published documents as the "experts". This was done to bypass the difficult, time consuming task of validation.

There is some justification for criticizing this approach. Any source of expertise should be reviewed and validated. However, the purpose of this prototype is to demonstrate that existing knowledge can be incorporated into ESs and provide help without a large development effort. Granted this method does not provide tailored knowledge to a particular program, but it does provide assistance to the untrained acquisition personnel presently on the job. As a follow on effort the tailoring of DoD manuals to specific programs would be the next logical step and in this stage validation will be very important.

G. PROJECTED USE, MAINTENANCE, AND SUPPORT

The projected use of this prototype is as a training aid until it can be incorporated into the entire ES. It is hoped that this prototype will prove useful as is. If nothing else, it provides another medium for disseminating the knowledge contained in these DoD manuals. It should serve as a good reference checklist or refresher for an experienced PM.

A copy of this prototype and thesis will be given to Department of Research and Information at the Defense Systems Management College (DSMC). There it can be evaluated with the other software development packages under development. After that, any further dissemination, and support will be determined by the DSMC.

V. SUMMARY

The Department of Defense (DoD) acquisition system has
been shown to be less than ideal in its ability to develop,
and produce new systems. One major cause of this has been
determined to be the lack of experienced personnel.
Furthermore, a continued inability to acquire working
defense systems will become a greater threat to the national
security of the United States. The lack of experienced
personnel suggests that a computer based Decision Support
System/Expert System (DSS/ES) could assist existing
personnel in developing the required expertise in a
shortened timeframe.

An examination of the definitions of these systems and
the problem definition was made. A good fit was found that
would allow for application of an ES. In order to allow for
a rapid development of the system, the problem space was
limited to one service and one type of equipment. The
problem space was also limited to the operational and
management control areas, due to the higher probability of
finding experts, and the greater stability found in those
areas.

```
/*  Main controls the entire program.  It allows the user to
specify the module of interest.  The module is then loaded
and executed.  The structure of each module is identical
except for the number of templates.  Upon completion, the
loaded module is deleted, leaving the main program ready to
execute again.   All rules are given a coded beginning
relating to its parent module.  The R and CR suffixes are
used to distinguish between rules and control rules.  All
rules are number in the same manner between modules.*/
/*BEGIN--main----------------*/
/*  Enable automatic question menu style*/
maincr-0:
     automaticmenu(ALL).
/*  Set the object that the system will seek for*/
mainr-0:
     goal = advice.
/*   Maincr-1 is the main execution statement for the
program.  Variables are requested in a set order in order to
determine program execution.  The capital letters indicate
variables that take on the name of used in the loaded
module.  This is unique to M.1 and the user should read the
M.1 technical manual before attempting to modify this.
Following rules support maincr-1.*/
maincr-1:
     if querry1 = Q1 and
          msg-Q1 = M and
          display(M) and
          querry2-Q1 = Q2 and
          msg-Q1-Q2 = M0 and
          display(M0) and
          exmsg-Q1-Q2 = M6 and
          quescont is sought and
          display(M6) and
          msg-ques1-Q2 = M1 and
          display(M1) and
          ques1-Q2 is known and
          ques1-Q2 = Q3 and
          msg-ques2-Q2 = M2 and
          display(M2) and
          ques2-Q2 is known and
          ques2-Q2 = Q4 and
          msg-ques3-Q2 = M3 and
          display(M3) and
          ques3-Q2 is known and
          ques3-Q2 = Q5 and
```

```
            msg-ques4-Q2 = M4 and
            display(M4) and
            ques4-Q2 is known and
            ques4-Q2 = Q6 and
            g(Q1,Q2,Q3,Q4,Q5,Q6) = Q0
        then advice = Q0.
/*   Supports maincr-1.   Prompts the user for the functional
area he is interested in.*/
maincr-3:
    question(querry1) = 'select the project area you want
advice on '.
/*   Provides list of possible answers.*/
mainr-1:
    l e g a l v a l s ( q u e r r y 1 )          =
[funding,design1,design2,test,production,logistics,
management,transition].
/*   Used to provide a manual pause to allow the user to read
the message displayed.   M.1 does not have an automatic pause
for displaying information,   therefore,  a question must be
used.*/
maincr-4:
    question(quescont) = 'Select "ready" to continue.'.
/*   Provides list of possible answers.*/
mainr-4:
    legalvals(quescont) = [ready].
/*   Main control rules 5 through 12 are used to find and
load the selected functional area code.*/
maincr-5:
    whenfound(querry1      =      funding)      =      [do(loadz
'b:funding.txt')].
maincr-6:
    whenfound(querry1      =      design1)      =      [do(loadz
'b:design1.txt')].
maincr-7:
    whenfound(querry1      =      design2)      =      [do(loadz
'b:design2.txt')].
maincr-8:
    whenfound(querry1 = test) = [do(loadz 'b:test.txt')].
maincr-9:
    whenfound(querry1      =      production)      =      [do(loadz
'b:product.txt')].
maincr-10:
    whenfound(querry1      =      logistics)      =      [do(loadz
'b:logisti.txt')].
maincr-11:
    whenfound(querry1      =      management)      =      [do(loadz
'b:managem.txt')].
maincr-12:
    whenfound(querry1      =      transition)      =      [do(loadz
'b:transit.txt')].
/*END----main----------------*/
```

```
/*   This ends the main section of the program.*/
/*    Transition  is  a  functional  area  consisting  of  one
template.   It  is  selected  because  of  this  fact.   Other
functional  area  with  more  than  one  template  operate  exactly
as  this  one.   The  naming  of  rules  follows  the  following
convention.   The  first  letter  designates  the  functional  area
that  is  t  for  transition.   The  second  letter  (  and  third  if
required  for  uniqueness)  designates  the  template,  that  is  t
for transition.*/
/*BEGIN--transition             section-------*/
/*------------transition section control----*/
/*    This  message  provides  the  user  with  the  list  of
templates he can choose from.*/
tcr-2:
    msg-transition   =   ['The   following   are   what   the
abbreviations stand for   ',nl,nl, 'transition = tt
                                            ',nl, '
                                                    ',nl,  '

  ',nl, '
            ',nl, '
                    ',nl, '
                            ',nl,
            '
            ',nl, '
                    ',nl, '
                            ',nl, '
                                    ',nl].
/*   Prompts the user to select a template.*/
tcr-3:
    question(querry2-transition)  =  'select  the  design  area
you want advice  on'.
/*   Provides list of possible answers.*/
tr-1:
    legalvals(querry2-transition)           =           [tt].
/*----------------transition-----*/
/*   The  first  message  provides  user  information  describing
the  template.   OVERVIEW  comes  from  the  NAVSO  P-6071  entry  at
the  beginning  of  each  template.   The  TIMELINE  comes  from  the
DoD  4245.7-M  entry  for  each  template.  REFERENCE  is  added
based upon the developers expertise.*/
ttcr-0:
    msg-transition-tt = ['OVERVIEW
    The  application  of  the  principles  briefly  discussed  in
the   templates   for   design,   test,   and   manufacturing   is
necessary   for   the   successful   accomplishment   of   the
engineering   tasks   on   schedule.   Integrated   with   and
pervading  this  effort  are  the  activities  presented  within
the   templates   for   facilities,   logistics,   and   management.
The   scope   and   interactions   for  this   multidisciplined
approach to risk reduction during development and production
```

are significant. A transition plan (DoD 4245.7-M) is necessary to identify the timing and application of the different disciplines, the risk-driving interrelationships, and particularly how and when execution of the plan is to be evaluated. To be effective the transition plan should be available at the start of engineering development and updated regularly until full production occurs.

TIMELINE

This effort begins prior to MS II and continues through the start of production. A transition plan, which is a comprehensive management plan describing all production-related activities that must be accomplished during design, test, and low rate initial production, is needed to ensure a smooth transition from development to full rate production. To be effective, the transition plan should be available before the start of FSD and updated regularly so that low rate production can be initiated at minimal risk.

REFERENCE ',nl,nl].
/* This message is used to provide the user with the textbook definition of the template. AREA OF RISK and OUTLINE FOR REDUCING RISK comes from the DoD 4245.7-M section in each template.*/
ttcr-5:
    exmsg-transition-tt = ['AREA OF RISK
In the past, a lack of formal transition planning has contributed significantly to the problems encountered in the transition from development to production. One of the major causes has been a Government/industry attitude that the performance parameters must be achieved during engineering development before expending funds to achieve production objectives. While there were a number of milestone-oriented Government requirements during the development phase and before the start of production, these were really stand-alone requirements generally used to verify the designs performance goals or as negotiation materials not having a smooth transition as an end objective.

OUTLINE FOR REDUCING RISK

1) Formal Government policies and specified contractual requirements that lay the groundwork for planning, programming, and executing specific actions during the development phase to ensure a smooth and successful transition to production are set forth in DoD Directive 4245.6 and DoD Directive 4245.7.

2) The Government program manager is required to fund and execute a contractor-developed transition plan, initially prepared no later that the start of engineering development and continually updated until rate production is achieved.

3) A sample transition plan outline includes, but is not limited to, consideration of all templates in this

85

Manual. The transition plan integrates the design, test, and manufacturing activities in order to reduce data requirements, duplication of effort, costs, and schedule. It identifies, for example, test and manufacturing issues that impact design, and design issues that affect test and manufacturing. The transition plan is a major means of implementing the manufacturing strategy described in one of the management templates.

    4) Development contracts contain the requirement for a formal design-to-unit production cost program and provisions for proof of manufacturing methods and processes. Funding is provided to the contractors for these areas of activity.

    5) Formal Production Readiness Reviews (PRRs) are conducted jointly by the customer and the contractor during the development effort and completed before the production decision. Participants in these reviews are qualified and experienced both in technical aspects of the product and the manufacturing processes proposed to produce it. PRRs, properly staffed and conducted, will result in both Government and contractor benefits. Government policy and procedures on conducting PRRs are contained in DoD Instruction 5000.38. ',nl,nl].

/* This next series of questions are the 4 traps form the NAVSO manual. Each trap is asked and the user is allowed to answer yes or no. The msg associated with each question is for providing extra explanation of the question. Presently, these are blank.*/

ttcr-1:
    question(ques1-tt)='Was the transition plan reviewed and approved by the Government at, just before, or after MS III?'.
ttr-1:
    legalvals(ques1-tt) = [yes,no].
ttcr-8:
    msg-ques1-tt =['',nl].
ttcr-2:
    question(ques2-tt)='Is the transition plan developed and reviewed only at the contractor program office level?'.
ttr-2:
    legalvals(ques2-tt) = [yes,no].
ttcr-9:
    msg-ques2-tt =['',nl].
ttcr-3:
    question(ques3-tt)='Is the transition plan only required in the contract and is not viewed as a corporate policy?'.
ttr-3:
    legalvals(ques3-tt) = [yes,no].
ttcr-10:
    msg-ques3-tt =['',nl].
ttcr-4:

```
        question(ques4-tt)='Is the Contractor planning for an 80
percent learning curve?'.
ttr-4:
        legalvals(ques4-tt) = [yes,no].
ttcr-11:
        msg-ques4-tt =['',nl].
/*----------------transition data list-----*/
/*   This portion contains the responses based upon the
answers given to the four questions.  The selection is based
upon  the  functional  area,  the  template  code,  and  the
yes/no/unknown responses.  Only one yes is allcwed in order
to  uniquely  get  a  response.   Unknowns  or  multiple  yes
responses will send the user to the reference section.*/
ttdr-1:
        g(transition,tt,no,no,no,no) = 'no traps found'.
ttdr-2:
        g(transition,tt,yes,no,no,no) = 'Trap #1 found
ALARM:   Contractor fails to generate and use the transition
plan prior to production start up.
CONSEQUENCES:  Much of the benefit of transition planning is
lost.
ESCAPES:    Contractor  should  prepare  and  use  a  transition
plan during early FSD.
BENEFITS:   All transition activities will be identified and
managed.'.
ttdr-3:
        g(transition,tt,no,yes,no,no) = 'Trap #2 found
ALARM:   Transition plan is developed only by the contractor
project office.
CONSEQUENCES:   Transition plan may be limited in scope.
ESCAPES:    Review  and  approve  transition  plan  at  corporate
level.
BENEFITS:  Corporate resources will be available to support
the transition plan. '.
ttdr-4:
        g(transition,tt,no,no,yes,no) = 'Trap #3 found
ALARM:   (1)  Manufacturing plan is presented as a transition
plan.
        (2)   Primarily  production  processes  and  equipment
are addressed by transition plan
CONSEQUENCES:  The government pays for a transition plan but
does not get one.
ESCAPES:   Reflect  an  integrated  corporate  strategy  in  the
transition plan:
                - Collocation of manufacturing and design team
                - Make or buy decisions
                - Capital investment considerations
                - Personnel recruiting and retention
BENEFITS:   Perturbations during production start up will be
minimized.  '.
ttdr-5:
```

```
     g(transition,tt,no,no,no,yes) = 'Trap #4 found
ALARM:    Contractor expects to achieve the 80 percent
learning curve by improving worker skills.
CONSEQUENCES:  Process is extremely slow and costly.
ESCAPES:   Contractor should define and fully implement a
transition plan.
BENEFITS:  Learning process will not be required. '.
ttdr-6:
     g(transition,tt,yes,yes,yes,yes) = 'all 4 traps found'.
ttdr-7:
     g(transition,tt,mm,mm,mm,mm) = 'You do not know much
about transition since you answered all of the questions
with unknown. Try using the reference portion of this
program to get to where you can answer the questions.'.
ttdr-8:
     g(transition,tt,ANY1,ANY2,ANY3,ANY4) = 'Can not tell.  A
combination of traps and/or unknown responses.  Please read
DoD 4245.7-M (pg 2-1) and NAVSO P-6071 (3-1) for further
information and help.'.
/*END----transition section----*/
/* After the end of each of the modules a short amount of
the main control section is present.  This remainder is
required to be here so that M.1 will not seek the default
responses first.  If the user does not respond in the
correct manner, ie, yes or no.  His response is converted to
mm here.  If this code is moved M.1 will find the default
first and not ask the user the questions.*/
mainr-5:
     if ques1-X is unknown
       then ques1-X = mm.
mainr-6:
     if ques2-X is unknown
       then ques2-X = mm.
mainr-7:
     if ques3-X is unknown
       then ques3-X = mm.
mainr-8:
     if ques4-X is unknown
       then ques4-X = mm.
/* In order for the program to be emptied a set of removal
code goes here  */
mainr-9:
     whenfound(advice)    =    [do(remove    tcr-2),do(remove
tcr-3),do(remove       tr-1),do(remove    ttcr-0),do(remove
ttcr-5),do(remove    ttcr-1),do(remove     ttr-1),do(remove
ttcr-8),do(remove     ttcr-2),do(remove    ttr-2),do(remove
ttcr-9),do(remove     ttcr-3),do(remove    ttr-3),do(remove
ttcr-10),do(remove     ttcr-4),do(remove   ttr-4),do(remove
ttcr-11),do(remove   ttdr-1),do(remove    ttdr-2),do(remove
ttdr-3),do(remove    ttdr-4),do(remove    ttdr-5),do(remove
ttdr-6),do(remove    ttdr-7),do(remove    ttdr-8),do(remove
```

```
mainr-5),do(remove     mainr-6),do(remove   mainr-7),do(remove
mainr-8),do(remove mainr-9)].
```

APPENDIX B
USER MANUAL

Installation, and Hardware Requirements

This prototype requires the M.1 system and the required hardware to execute it. Please see the M.1 technical manual for this information. The only peculiar installation for this prototype is that rules in the main file "maincr-5" through "maincr-12" must reflect the correct directory or drive in order to find the modules. According to the user's desires the modules can be loaded into any drive or directory as long as the above rules are changed to reflect the correct location.

The user modifies these rules by entering M.1 and loading the file named "PROGT1.FST" (See below section). After loading this file press the "F10" key to enter the menu panels. Using the cursors move to the menu panel named "Knowledge Base" (second form the left), and move down to edit an entry. Press enter and M.1 will ask for a rule name. Type in "MAINCR-5" and return. M.1 will then call up this rule and display it. Move the cursor to the appropriate section and replace the drive or directory. CAUTION, M.1 is particular about the single quotes ''. Only use the single quote to begin and end the location. DO NOT CHANGE ANY brackets or parenthesis. Upon completion of the

change, press "F10" to enter the change. Repeat this until rules 5 through 12 are correct.

Upon completion of the changes, press "F10" again and go to the knowledge base panel. Select the save kb in fast format is highlighted, press enter. M.1 will prompt you for a file name. At this time enter single quote, drive letter, colon, directory information, and the main program file name "PROGT1.FST", single quote. After checking this, press enter and M.1 will save the file as modified. If the quotes are not used M.1 will save the file to the default drive and directory. This is not serious but is very scary and annoying. At this time the program is ready to execute.

It is never a good idea to load application files in the same directory or disk drive as the program files. Therefore, this manual assumes that the user has loaded this prototype into a different directory or disk drive.

Operation

This program requires the M.1 system to be installed. The user starts the M.1 program by either typing "M1" , invoking an already installed autoexecution file, or via a menu selection program. Once M.1 is running the user must load the main program file. This is accomplished by pressing the alt key and "L" simultaneously. M.1 will read the default drive and directory and display the file names. At this point M.1 will allow the user to press the "F2" key

for an alternate directory or drive. In the same provided type the directory or disk drive that the prototype software is loaded in. After pressing "RET", M.1 will read the newly designated directory or drive. The filenames will appear with the first file highlighted. By using the cursor keys, move the highlighting to the file named "PROGT1.FST", press enter and M.1 will proceed to load the file. While loading a loading sign will flash in the lower right hand corner of the screen. Upon completion, this sign will return to a non-flashing ready.

The user is now ready to begin the consultation. By pressing the alt key and the "G" key simultaneously, M.1 will begin executing the program. The program will prompt the user for the functional area of interest. Selection is made via the cursor keys and pressing return. Upon selection, M.1 will proceed to locate and load the functional module code. The user may now respond to the questions in the appropriate manner. WARNING M.1 allows for the use of "unknown" responses. This prototype will trap those responses but not give the user any useful advice. A feature for providing a reference section for unknown responses is being worked on.

After completing the four questions, the program will return a ready sign in the lower right hand corner. This signifies that the session is complete. M.1 still has the main program module loaded in its rule base. This allows

the user to restart the prototype by merely pressing alt key and "G" again. However, if any of the larger modules have been executed, M.1 will have insufficient memory to allow another large module to be run. This is due to the fact that the variables form the previous module are not zeroed out. Therefore, if the user attempts to execute another module M.1 may issue a memory error and return to DOS. To date the only way found to avoid this is to exit M.1 and reenter it.

A final note, the entire command structure of M.1 is enabled during the consultation. Any valid M.1 commands may be issued. In particular the scroll function command "F2" is necessary to read certain of the screens. Upon reading the user presses the esc key and M.1 resumes operation.

LIST OF REFERENCES

Alavi, M. and Napier, H. A. 1984. "An Experiment in Applying the Adaptive Design Approach to DSS Development". Information and Management, vol. 7, no. 1. Reprinted in **Decision Support Systems**. Ralph Sprague and Hugh Watson, [eds.]. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

Bennett, J. 1977. "User-Oriented Graphics, Systems for Decision Support in Unstructured Tasks". In **User-Oriented Design of Interactive Graphics Systems**, S. Treu [ed.]. New York: Association for Computing Machinery.

Defense Systems Management College. 1986. Technical Report, "Concept For An Extension of the Acquisition Expert System to Other Phases of the Acquisition Cycle". Unpublished technical report prepared under contract by The Analytic Sciences Corporation.

Drake, R. G. and Minnema, J. E. 1988. "Expert System/Decision Support System for Department of Navy Automatic Data Processing Equipment Acquisition". Unpublished working paper, Monterey, California: US Naval Postgraduate School.

Gorry, G. Anthony, and Scott Morton, Michael S. 1971. "A Framework for Management Information Systems". **Sloan Management Review**, vol. 13, no. 1. Fall.

Goul, Micheal, and Tongue, F. 1987. "Project IPMA: Applying Decision Support System Design Principles to Building Expert-Based Systems". **Decision Sciences**, 18.

Herres, R. T. 1983. "Interview". **Defense Science & Electronics**, September.

Hogue, J. T., and Watson, Hugh J. 1984. "Current Practices in the Development of Decision Support Systems". Proceeding from the Int. Conf. on Information Systems. Reprinted in **Decision Support Systems**, Ralph Sprague and Hugh Watson, [eds.]. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

O'Leary, D. E. 1987. "Validation of Expert Systems - With Applications to Auditing and Accounting Expert Systems". **Decision Sciences**, 18.

President's Blue Ribbon Commission on Defense Management. 1986. **A Report to the President on Defense Acquisition**. Washington, D.C., April.

Pressman, R. S. 1987. **Software Engineering--A Practitioner's Approach**. 2nd ed. New York: McGraw-Hill.

Smith, H. 1988. **The Power Game**. New York: Random House.

Sprague, R. H. Jr. 1980. "A Framework for the Development of Decision Support Systems". **MIS Quarterly**, 4(4). June. Reprinted in **Decision Support Systems**. Ralph Sprague and Hugh Watson, [eds.]. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

Sprague Ralph H., and Carlson, Eric D. 1982. **Building Effective Decision Support Systems**. Englewood Cliffs, New Jersey: Prentice-Hall.

Stephanou, S. E., and Obradovitch, M. M. 1985. **Project Management - System Development and Productivity**. Malibu, California: Daniel Spencer Publishers.

Turban, E., and Watkins, P. R. 1985. "Integrating Expert Systems and Decision Support Systems". In **Transactions of the Fifth International Conference on DSS (DSS '85)**. Reprinted in **Decision Support Systems**. Ralph Sprague and Hugh Watson, [eds.]. Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

U.S. Congress. Senate. Armed Services Committee. 1985. Staff Report, **Defense Organization: The Need For Change**. S. Prt 99-86, 99th Cong., 1st sess. Washington, D.C.:Government Printing Office.

U.S. Department of Defense. 1985. **Manual DoD4245.7-M, Transition From Development to Production**. Published by the Assistant Secretary of Defense for Acquisition and Logistics. Washington, D.C.

U.S. Department of the Navy. 1986. **NAVSO P-6071, Best Practices - How to Avoid Surprises in the World's Most Complicated Technical Process.** Published by the Chairman, Defense Science Board Task Force, March.

U.S. Department of the Navy. 1987. **Navy Program Manager's Guide,** 1987 Edition - For Comment Draft. Prepared by William C. Ruckert under Purchase Order No. N66001-86-M-B482.

Washington Post. 1985. "Defense INC.". 31 March - 15 December.

Waterman, D. A. 1986. **A Guide to Expert Systems.** Reading, Massachusettes: Addison-Wesley Publishing Co.

Wolfgram, D. D., Dear, T. J., Galbraith, C. S. 1987. **Expert Systems for the Technical Professional,** John Wiley & Sons, New York, New York.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center   2
    Cameron Station
    Alexandria, VA 22304-6145

2. Library, Code 0142   2
    Naval Postgraduate School
    Monterey, CA 93943-5002

3. Superintendent, Code 74   1
    Attn: Prof. C. R. Jones
    Naval Postgraduate School
    Monterey, CA 93943-5000

4. Superintendent, Code 54HR   1
    Attn: LCDR E. N. Hart
    Naval Postgraduate School
    Monterey, CA 93943-5000

5. Captain James E Minnema, USMC   2
    147 Bayberry Road
    Newport, NC 28570

6. Director, PMSS Directorate (DRI-S)   2
    Attn: Mr. Harold J. Schutt
    Defense Systems Management College
    Fort Belvoir, VA 22060-5426

7. Commandant of the Marine Corps   1
    Code TE 06
    Headquarters, U.S. Marine Corps
    Washington, D.C. 20360-0001

8. Superintendent, Code 55WO   1
    Attn: Prof. M. Woods
    Naval Postgraduate School
    Monterey, CA 93943-5000

9. Assistant Secretary of the Navy   1
    (Shipbuilding and Logisitics)
    Department of the Navy
    The Pentagon
    Washington, D.C. 20350

10. Under Secretary of Defense for Acquisition   1
    The Pentagon
    Washington, D.C. 20301-1155

11. Assistant Secretary of Defense   1
    (Command, Control, Communications, and Intelligence)
    The Pentagon
    Washington, D.C. 20301-1155

12. Commanding General, Marine Corps Research,   1
    Development, and Acquisition Command
    Attn: Deputy for Programs
    Quantico, VA 22134-5080