NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN EXPERT SYSTEMS APPROACH TO
MILITARY DECISION SUPPORT IN AN
AIR DEFENSE SCENARIO

by

Stephen P. Dodd
and
Michael J. Molidor

March 1987

Thesis Advisor: T. R. Sivasankaran

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT: 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): 54

7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School

6c ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000

7b ADDRESS (City, State, and ZIP Code): Monterey, California 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 NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification): AN EXPERT SYSTEMS APPROACH TO MILITARY DECISION SUPPORT IN AN AIR DEFENSE SCENARIO

12 PERSONAL AUTHOR(S): Dodd, Stephen P. and Molidor, Michael J.

13a TYPE OF REPORT: Master's Thesis

13b TIME COVERED: FROM ________ TO ______

14 DATE OF REPORT (Year, Month, Day): 1987 March

15 PAGE COUNT: 97

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): expert system; artificial intelligence systems; air defense scenario; stochastic non-repetitive environment

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

An expert system is a subspecialty of various artificial intelligence systems which has relevance for military applications. This thesis is concerned with examining the feasibility of using an expert systems approach for solving problems in a stochastic non-repetitive environment. To ascertain this feasibility, the expert system prototype supporting an air defense scenario was developed. Additionally, the prototype was evaluated to see if it was possible to interface sources external and independent of the expert system to provide the required inputs for processing and analysis.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL: Prof. T.R. Sivasankaran

22b TELEPHONE (Include Area Code): (408) 646-2637

22c OFFICE SYMBOL: Code 54Sj

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

An Expert Systems Approach To Military Decision Support In An Air Defense Scenario

by

Stephen P. Dodd
Captain, United States Marine Corps
B.S., Texas A&M University, 1977
and
Michael J. Molidor
Captain, United States Marine Corps
B.S., Texas A&M University, 1978

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN INFORMATION SYSTEMS
and
MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Command, Control and Communications)
from the
NAVAL POSTGRADUATE SCHOOL
March 1987

ABSTRACT

An expert system is a subspecialty of various artificial intelligence systems which has relevance for military applications. This thesis is concerned with examining the feasibility of using an expert systems approach for solving problems in a stochastic non-repetitive environment. To ascertain this feasibility, the expert system prototype supporting an air defense scenario was developed. Additionally, the prototype was evaluated to see if it was possible to interface sources external and independent of the expert system to provide the required inputs for processing and analysis.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. OVERVIEW

An expert system is a subspecialty in the field of artificial intelligence which has relevance for use in a military environment. Significant advances have been made in the area of expert systems, and a host of military applications are expected. [Ref. 1: p. 29] Expert systems are computer programs that duplicate to a certain degree the kinds of conclusions and results that an expert would make given the facts and circumstances. Expert systems are designed to solve problems, to give advice, and to provide a rationale for the decision reached.

Although there have been some impressive results demonstrated by expert systems and several commercial examples exist, the most well known expert systems have been limited to applications in medical diagnosis, mineral prospecting, chemical analysis and computer configuration. The developers of these commercial systems make no pretext that these expert systems duplicate the reasoning process that experts commonly use in problem solving; instead they have concentrated on achieving results similar to those of an expert. [Ref. 1: p. 36] Current applications of expert systems have been limited to non-military environments. The aim of this thesis is to apply the expert systems approach to a stochastic non-repetitive problem, specifically, a simulated air defense scenario.

B. RELEVANCE OF EXPERT SYSTEMS TO THE MILITARY

The air defense mission for U.S. Military Forces is characterized by a requirement for calculated, rapid and aggressive decision making in a non-structured environment. The decision to employ a particular weapon system to thwart a projected threat must be based upon a knowledge of assets available and some type of information on the enemy. The multitude of variables encapsulated within the decision making requirement and the necessity for sudden immediate responses to potential threats forces the weapons employment commander to be thoroughly familiar with all facets of the air defense environment. A delay in countering an enemy action can be as deadly as a misapplication of friendly forces. The task of providing air defense is becoming increasingly more difficult in todays high technology era. Sophisticated enemy weapon platforms can penetrate a surveillance zone and strike at a vital area in a matter of minutes. A complex organization of defense assets exists within the military command and control structure to combat these platforms. However, the means to circumvent or mininize losses is predicated upon the commanders knowledge and ability to manage all available resources. The commander is in effect tasked to be an 'expert' in the decision support arena.

In any high stress, high information rate situation, human decision makers can become overloaded and unable to assimilate and use all of the data that is available to them. This is particularly true in sensor interpretation and combat management where data received must be evaluated and used effectively in managing weapons and making tactical decisions under the auspices of time constraints. Expert systems offer the potential for application in decision support by providing interpretation aids, automatic analysis or situation assessment. In these applications, expert systems would automatically assimilate the latest data from such sources as sensors, intelligence reports and human decision makers, and could provide plausible hypothesis about current situations to operators, planners, and other human decision makers. [Ref. 1: p. 42]

Expert systems can best be used in critical military tasks where the development of new machine technologies will aid in the use and application of strategic computing. Some of the goals of strategic computing include: [Ref. 2: p. 82]

- Enable the operation of military systems under critical constraints such as time, information, and overload
- Enable the management of forces/resources under constraints of information overload, geographic distribution, and cost of operations
- Facilitate the design, manufacture and maintenance of defense systems within time, performance, quality, reliability and cost constraints

Utilizing expert systems in a strategic computer environment will allow the military commander to exercise judgements and decisions more effectively if more information is able to be processed for interpretation, which is especially critical when dealing with constraints such as time information overload. Expert systems might also enable military personnel to more efficiently and effectively manage the forces and/or resources that may be involved in a scenario. Expert systems can process the information presented under multiple parameters and constraints, putting forth a recommended course of action to take. This would give military personnel a fast, responsive course of action that may be taken based upon the operating environment. Using expert systems in this manner might allow military personnel to improve their skills, knowledge, and training in different environments.

C. THESIS RESEARCH QUESTIONS

The primary goal and emphasis of this thesis is to determine the feasibility of using the expert systems approach to solve a problem in a stochastic non-repetitive environment. To accomplish this goal we needed to research and answer two primary questions:

1. Can an expert systems approach be applied to problems that are characterized by a non-repetitive stochastic environment?
2. Can a developed expert system prototype be interfaced with an external system in order to draw the required inputs for processing and analysis?

The analysis, answers, and conclusions to these questions will enable us to determine the viability of any future extensions of our thesis where the incorporation of actual sensor information could be used directly as input to an expert system.

D. DEVELOPMENT OF AN EXPERT SYSTEMS PROTOTYPE

The expert systems approach used in this thesis is based upon multiple parametric representations of activities that may transpire in a stochastic non-repetitive environment. The circumstances and information encountered within this environment will have to be interpreted by the expert system prototype constructed. The facts and information representing the environment will be developed and presented through data base files.

The prototype is built on the premise that it will use information about a developing situation represented in data base files, and will ascertain a correct course of action that should be taken in response to the scenario presented. The non-repetitive variables to be used by the system is data that would actually be gathered by sensors in a real world scenario. To replicate different types of situations, database files will be constructed that will 'feed' the required inputs to the expert system prototype for processing, analysis, and decision recommendation. In addition to data files stored within the expert system prototype, a database file will also be built using DBASE III Plus in order to test the use and application of an external system interfacing with an expert system prototype. By using an external system such as DBASE III Plus, an attempt will be made to ascertain whether or not it is feasible that expert systems which might be applicable to military environments can be integrated and used with computer systems external and independent of an expert system.

In order to develop an expert system prototype for use in this thesis the *Arity/Expert Development Shell* as well as Prolog were utilized. The *Arity/Expert*

Development Shell is a collection of developmental tools which provided the basis for constructing an expert system prototype. The power of the shell was enhanced by interleaving it with a Prolog interpreter.

II. CONSTRUCTION OF AN EXPERT SYSTEM

A. ARTIFICIAL INTELLIGENCE

Artificial Intelligence is a growing set of computer problem-solving techniques that are developed to imitate human decision-making processes. There are many definitions of artificial intelligence (AI) put forth by AI experts, some of which include:

- Marvin Minsky, Donner Professor of Science at the Massachusetts Institute of Technology and founder of the field of AI, has stated that "artificial intelligence is the science of making machines do things that would require intelligence if done by men." [Ref. 3: p. 4]
- Patrick H. Winston, director of the artificial intelligence laboratory at the Massachusetts Institute of technology, writes: "The goals of artificial intelligence are to make computers more useful and to understand the principles which make intelligence possible." [Ref. 4: p. 1]
- Nils Nilson, Chairman of the Department of Computer Science at Stanford University, says: "The field of Artificial Intelligence has as its main tenant that there are indeed common processes that underlie thinking and perceiving, and furthermore that those processes can be understood and studied scientifically." [Ref. 5: p. 8]
- Bruce G. Buchanan, adjunct professor of computer science research; and Edward A. Feigenbaum, principal investigator at the Heuristic Programming Project; both of Stanford University write: "Artificial Intelligence research is that part of computer science that investigates symbolic, nonalgorithmic reasoning processes and the representation of symbolic knowledge for use in machine inference." [Ref. 6: p. 15]

The field of artificial intelligence offers distinctive approaches to complex problem solving. In conventional computing, it is the programmer who creates computer program instructions that follow solution paths for each situation presented. The solution path is planned by the programmer using structured predictable steps, which enable the solving of problems that require a large amount of data to be processed. Artificial intelligence uses techniques that are helpful in solving complex problem-symbolic processing. The symbols processed by artificial intelligence programs represent real-world entities, and instead of simply performing calculations, artificial intelligence programs manipulate the relationships among the symbols. [Ref. 5: p. 52]

B. EXPERT SYSTEMS: AN OVERVIEW

The area of expert systems investigates methods and techniques for the construction of man-machine systems that have specialized decision-making expertise. Expertise consists of knowledge about a particular domain, understanding of the domain problem, and skills at solving the problems. Knowledge in any specialty can be gained from two areas: public and private. Public knowledge involves the published definitions, facts and theories of which textbooks and references in the discipline of study are typically composed. Expertise usually involves more than just public knowledge, human experts generally possess private knowledge that has not found its way into published literature. This private knowledge consists largely of rules of thumb used by experts, commonly known as heuristics. Heuristics enable the human expert to make educated guesses when necessary, to recognize promising approaches to problems, and to deal effectively with incomplete or erroneous data. Elucidating and reproducing such knowledge is the central task in constructing expert systems. [Ref. 7: p. 4] Expert systems differ from both conventional data processing systems and systems developed in other branches of artificial intelligence. AI applications generally involve several distinguishing features, which include symbolic representation, heuristic search, and symbolic inference. Expert systems also differ from the broad range of AI in several aspects. First, they emphasize domain-specific problem solving strategies; Second, they employ self-knowledge to reason about their own inference processes and provide explanations or justifications for the conclusions reached; Third, they solve problems that generally fall into one of the following categories: interpretation, prediction, diagnosis, debugging, design, planning, monitoring, repair, instruction, or control. As a result of these distinctions, expert systems represent an area of AI research that involves paradigms, tools, and system development strategies. [Ref. 7: p. 52]

Using the above parameters an expert system can be defined as a knowledge-intensive program that solves problems that normally require a human expert. An expert system performs many of the reasoning functions an expert does such as asking relevant questions and explaining its reasoning. According to Frederick Hayes-Roth, characteristics common to all expert systems are: [Ref. 8: p. 264]

- They reason heuristically, using what experts consider effective rules of thumb
- They interact with humans in appropriate ways, including the use of natural languages
- They manipulate and reason about symbolic descriptions
- They function with erroneous data and uncertain judgemental rules
- They contemplate multiple competing hypothesis simultaneously
- They explain why they are asking questions
- They justify their conclusions

Compared to a human expert today's expert systems appear narrow, brittle, and shallow lacking a human experts breadth of knowledge and understanding of fundamental principles. Expert systems today replicate the human decision-making process rather grossly. They make major decisions by elucidating many of the relevant criteria and by making many of the educated guesses that human experts might make if forced to verbalize their thought processes. Additionally, today's expert systems do not learn from experience. [Ref. 8: p. 264]

C. BASIC CONCEPTS OF PROBLEM SOLVING

The basic concepts of intelligently approaching problem-solving solutions shown in Figure 2.1 are the ideas that motivate and explain the aspects of knowledge resident in expert systems. The central notion of intelligent problem-solving is that a system must construct its solutions selectively and efficiently for given alternatives.. When the expert is resource limited, he needs to search his alternatives in order to achieve high performance by using knowledge acquired to effectively and efficiently solve his problem. [Ref. 7: p. 19] In most cases, experts face problems that are not easily formalized or do not have algorithmic solutions, and normally, heuristic methods must be used. Hence, effective solutions depend on the timely use of knowledge to identify potential decisions that are promising and rule out unpromising ones. The first three concepts in Figure 2.1 address how an expert can develop the primacy of knowledge that must be resident in an expert system. The fourth and fifth concepts are methods that the expert can follow to aid in developing a more efficient knowledge base and to increase the ability of the system to solve more difficult problems.

D. KNOWLEDGE ENGINEERING

Knowledge engineering is the term that has been adopted by researchers for the discipline of formulating knowledge in expert systems. The burden of uncovering and formalizing the experts' knowledge is incumbent upon the knowledge engineer. Through an extended series of interactions with an expert, the knowledge engineer should be able to define the problems to be attacked, discover the basic concepts involved, and develop rules that express the relationships that exist between these basic concepts. The term knowledge engineering combines scientific, technological and methodology elements. One of the principles of knowledge engineering is that expert performance rarely conforms to some algorithmic process. The essential tasks of knowledge engineering is extracting, articulating, and computerizing the experts knowledge.

1. Knowledge = Facts + Beliefs + Heuristics
2. Success = Finding a good enough answer with the resources available
3. Search efficiency directly effects success
4. Aids to Efficiency:
    a. applicable, correct, and discriminating knowledge
    b. rapid elimination of "blind alleys"
    c. elimination of redundant computation
    d. increased speed of computer operation
    e. multiple, cooperative sources of knowledge
    f. reasoning at varying levels of abstraction
5. Sources of increased problem difficulty
    a. erroneous data or knowledge
    b. dynamically changing data
    c. the number of possibilities to evaluate
    d. complex procedures for ruling out possibilities

Figure 2.1 Basic Concepts of Problem Solving.

Knowledge in the abstract consists of descriptions, which are known as relationships, and procedures in some basic domain. A knowledge base contains particular descriptions which are known as relationships. Knowledge is what enables a human expert to solve difficult problems in a precise manner, the solution to which may be determined in many forms: empirically, heuristically, and causality. Knowledge engineering addresses the problem of developing and building skilled computer systems which first extract the experts knowledge, then organize it in an effective implementation. [Ref. 7: p. 13]

E. TYPES OF EXPERT SYSTEMS

Most expert systems fall into one of several generic types, each of which will be briefly discussed herein. Interpretation systems infer situation descriptions from observations, which includes surveillance, speech understanding, image analysis, chemical structure elucidation, signal interpretation, and different types of intelligence analysis. An interpretation system explains stochastic data by assigning symbolic meanings describing the situation or system state accounting for the data.

Prediction systems infer likely consequences from given situations. This category includes weather forecasting, demographic predictions, traffic predictions, crop estimations, and military forecasting. A prediction system typically employs a parametric dynamic model with parameter values that are fitted to the given situation.

Diagnosis systems infer system malfunctions from observables. This category includes medical, electronic, mechanical, and software diagnosis. Diagnosis systems typically relate observed behavioral irregularities with underlying causes using one of two techniques. The first method uses a table of associations between behaviors and diagnoses. The second method combines knowledge of system design with knowledge of potential flaws in design, implementation, or components which generate candidate malfunctions that are consistent with the observations.

Design systems construct descriptions of objects that will satisfy the constraints of the design problem. Such problems include circuit layout, building design, and budgeting. Design systems construct descriptions of objects that are in various relationship with one another and that verify that the configurations conform to stated constraints.

Planning systems design actions. These type of systems specialize in problems of design concerned with objects that perform functions. They include automatic programming as well as robot, project, route, communication, and military planning problems.

Monitoring systems compare observations of system behavior to features that are crucial to successful plan outcomes. These crucial features correspond to potential flaws that may be present in the plan. Generally, monitoring systems identify vulnerabilities in two ways. The first type of vulnerability corresponds to an assumed condition whose violation would nullify the plan's rationale. The second type of vulnerability arises when some potential effect of the plan violates a planning constraint. Many computer-aided monitoring systems exist for nuclear power plants, air traffic, disease, and regulatory fiscal management tasks.

Debugging systems prescribe remedies for malfunctions. These systems rely on planning, design, and prediction capabilities to create specifications or recommendations that will help correct a diagnosed problem. Computer-aided debugging systems are available in the form of intelligent knowledge base and text editors, although none yet qualify as an expert system.

Repair systems develop and execute plans that administer a remedy for some diagnosed problems. Some of these systems incorporate debugging, planning, and execution capabilities. Computer-aided systems are occuring in the domains of automotive, network, avionic, and computer maintenance, however, expert systems are just now entering this domain.

Instruction systems debug and diagnose behaviors. They incorporate diagnosis and debugging subsystems that specifically address certain problems. These systems begin by constructing a hypothetical description of the knowledge pertaining to the problem, then diagnose weakness in the knowledge and identify an appropriate remedy.

The last type of system is called control. An expert control system adaptively governs the overall behavior of a particular system. In order to accomplish this a control system must repeatedly interpret the current situation, predict the future, diagnose the causes of anticipated problems, formulate a remedial plan, and monitor its execution to ensure success has been achieved. Problems addressed by control systems include air traffic control, business management, mission control and battlefield management. [Ref. 7: pp. 13-16]

F. A FRAMEWORK FOR EXPERT SYSTEMS

One of the most important aspects of an expert system is that it can make decisions or help people make decisions. Using this basic capability of an expert system, the essential framework behind expert systems can be illustrated. One of the characteristics of an expert system is that emphasis is on qualitative logical reasoning, vice quantitative calculations that are used in most other type of programs. Logical inference uses logical data, and consequently, the database of an expert system may or may not contain numerical data. The logical data in an expert system is commonly known as the knowledge base.

One of the key features to a knowledge base is that its contents are not just abstract symbols like numbers or conditional probabilities. Another important component of a knowledge base is a body of facts which represents the way in which concepts are related to each other. In expert systems, the aim is to represent meaning explicitly by recording the relationship in a form that the computer can exploit, and not by reducing data to abstract quantities. [Ref. 9: p. 129]

Figure 2.2 is a schematic representation of the many elements found in an expert system. [Ref. 9: p, 131] The most primitive element shared by all other computer

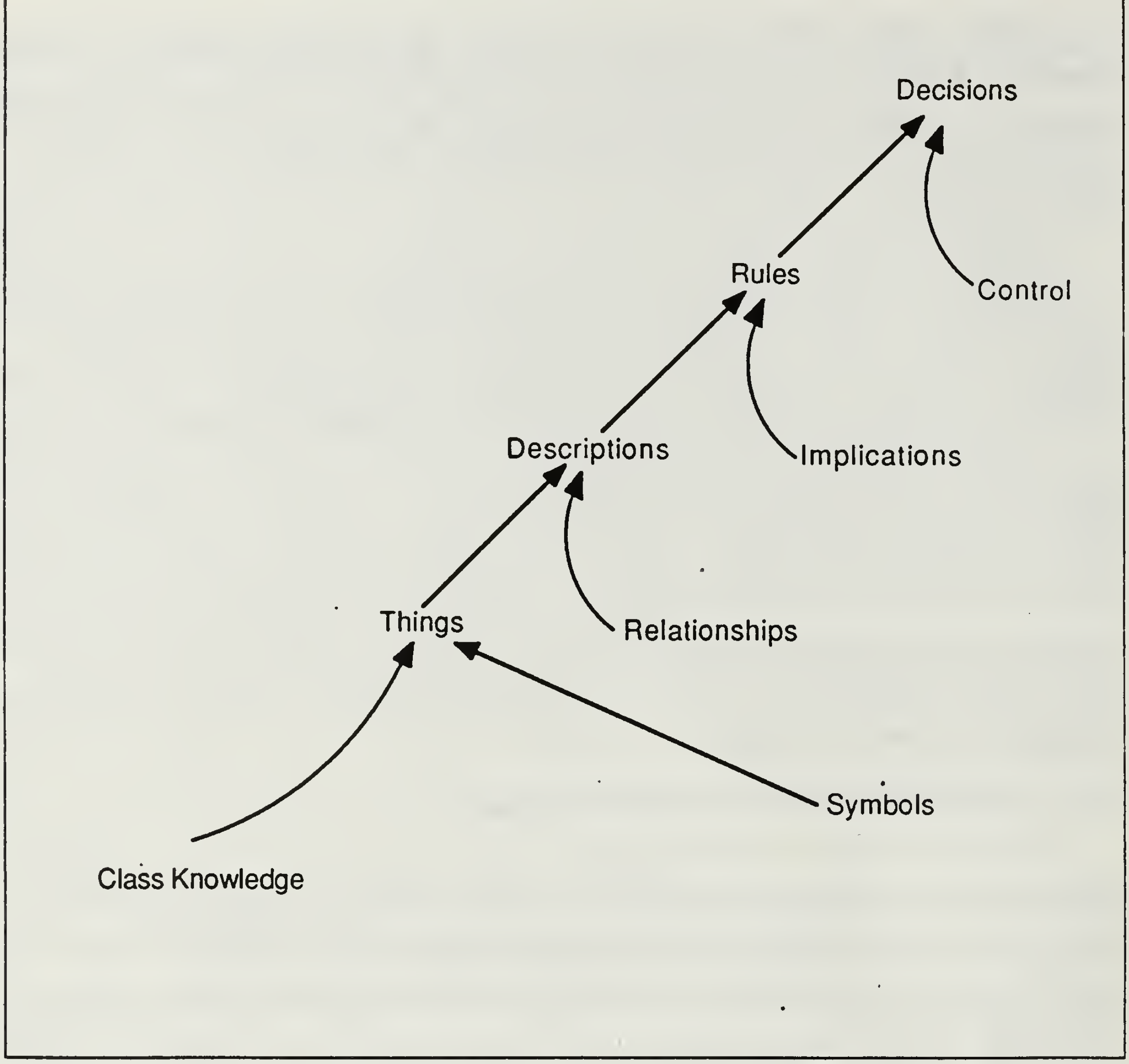

Figure 2.2 Schematic Representation of an Expert System.

programs is the symbol. Symbols are simply character strings which stand for or suggest something by reason of relationship, association, convention or accidental resemblance. When elements are progressively added, the ability of these symbols becomes richer and much more useful. After the descriptions and relationships have been instantiated, the rules of the knowledge base are constructed. Given data, the rules should be capable of making inferences under the auspices of some type of control process. A skillful decision maker does not apply rules blindly to a situation, but rather in a controlled manner, and an expert system must be similary controlled.

The inference engine of a knowledge based decision process usually provides this control.

The inference engine in a knowledge based system uses information presented in the knowledge base to infer conclusions. The inference engine interprets the rules in the base in conjunction with the control options in order to infer solutions to consultations. There are two main ways in which inference engines control the use of knowledge in solving problems. In the 'data-driven' method the inference engine examines the data that is available and applies all the rules which are satisfied to derive the new data. The conclusions which are made contribute to the body of data and may in turn allow other rules to operate. This cycle may be repeated many times until eventually, a definite diagnosis is reached or a recommendation is concluded. The second method, the 'goal-driven' approach begins at the opposite end of the logical process. The goal-driven approach identifies the conclusions that it needs to establish its goals, and then examines the rules to see which ones contribute to these goals. The next step is to look at the "If. . . " part of the "If. . . Then" rules and locate any missing data that may be needed to reach the conclusions. Missing data is then filled in as new 'subgoals' or through an interactive process with the user. As with the data-driven method, the cycle may be repeated many times, and it may require many generations of subgoals before the expert system is able to obtain all the necessary data to achieve its goals. If the facts and rules are sufficiently comprehensive, and they contain all the information that is logically required to make a decision, then all the inference engine has to do is make sure all the relevant facts and rules are retrieved at the same time. If the facts and rules are not sufficiently comprehensive then the inference engine will be unable to make conclusions based on the facts presented. The power of an expert system comes from the quality and comprehensiveness of the knowledge base not from sophisticated features of the system. [Ref. 9: p. 131] The details and construction of an expert system vary enormously from design to design, albeit, the idea of knowledge is always central.

G. STEPS IN CONSTRUCTING AN EXPERT SYSTEM

In constructing an expert system the goal is to replicate expert decision making with the aid of a computer. In order to accomplish this requires a well planned and flexible design strategy. The phases involved in constructing an expert system are:

1. Determine if a task is suited to expert system technology
2. Knowledge acquisition

3. Implementation Stage
4. Initial system design
5. Building a prototype system
6. Analysis and redesign
7. Final analysis and fine tuning
8. System maintenance

1. Determining if a task is suited to expert system technology

One of the major reasons for the development of expert systems was for applications and solutions to problems with situations where a programming algorithm would not be practical. An analysis of the task to be performed should be accomplished to ensure that the expert systems approach is the best and most practical way to address the problem, or perhaps the problem is best suited for some other type of programming technique.

There are several characteristics that are commonly associated with the use of expert system technology:

- Tasks which involve knowledge that can be expressed as "rules of thumb"
- Problems which human experts typically perform
- Tasks which involve knowledge that is rapidly changing
- Tasks which do not rely heavily on "common sense"
- Tasks which involve inexact reasoning
- Tasks that can be represented as a set of independent actions or conditions

These tasks represent areas that might be useful for the application of expert system technology.

2. Knowledge acquisition

This next phase involves the gathering of information and formalization of data that will be necessary to develop an expert system, this includes the identification of all the parameters, including participants, problem characteristics, resources and goals, which are associated with tasks or problems under consideration. From the acquisition and analysis of knowledge comes the entire development strategy of how the follow-on phases will be carried out. During this phase the major components of the reasoning process and identification of the elements that should be included in the expert system are determined.

Refinement of the acquired knowledge and organization of key elements has to be accomplished before any further system development. The analysis of the knowledge and elements should be according to the interrelationships between the elements. Instances where two or more reasoning paths make use of the same information, and identification of how these reasoning paths are related should be noted and identified. This will allow the developers of the expert system to continually refine the knowledge acquired and help eliminate unnecessary or redundant information. The concepts and relations that are identified in the first phase are made explicit during this phase. Questions such as: What type of data is available?, What is given and inferred?, and, What processes and constraints are involved in the problem solution?, are asked by the knowledge engineer to the source of his knowledge. There has to be repeated interactions between the knowledge engineer and the expert source in order to develop key concepts and relations.

3. **Implementation Stage**

Implementation involves the mapping of the formalized knowledge into the represented framework associated with the tools to solve the problem. As the knowledge in this framework is made compatible, consistent and organized, the framework becomes an executable program. The knowledge engineer then evolves a useful representation for the knowledge collected and will use it to develop a prototype expert system. The development of the prototype system is an extremely important step in the construction of an expert system. The prototype knowledge base is implemented by using whatever knowledge engineering aids available and chosen for development of the program. The formalization involves mapping the key concepts, subproblems and information flow characteristics that have been isolated during the knowledge acquisition phase, into formal representations based upon knowledge engineering tools or frameworks. There are two important factors during this stage: the underlying model of the process and the characteristics of the data. Formalization of these two factors with the conceptual information flow and the subproblem elements organizes the framework and the sketch of the concepts and relations inherent in the problem to be solved.

4. **Initial system design**

This next phase involves the organization of information acquired during the knowledge acquisition phase and modelled in the implementation stage, into a form that will be usable by the expert system. The overall purpose of this expert system should be kept in mind when designing the system. The design should reflect the most efficient manner in which to represent the acquired knowledge and should also provide a user interface that will be appropriate for users of the system who may not fully understand expert systems. In system design, representation of the knowledge needs to be understood, and a determination needs to be made as to the best way to represent the knowledge that will be used by the system. The expert system makes use of a taxonomy file and a rule base file to store the knowledge that has been acquired and refined by the experts.

5. **Building a prototype system**

After the initial system design has been finalized, the next phase is to build a prototype of the expert system. This phase incorporates the following steps:

1. The initial taxonomy and rules are constructed
2. Production of an interpreter that incorporates the elements of the taxonomy
3. Validity of the taxonomy and rule base is tested through the use of expert system predicates

The development of the prototype system should be accomplished incrementally. Limited versions of the rule base file and the taxonomy file should be coded and tested. After these initial rules and taxonomies have been tested then further construction of these two files can be accomplished and incorporated with the original files. The finished prototype should be a complete working system, and from this prototype the expert should be able to determine if the system is performing as an expert would when presented with the circumstances and problems.

6. **Analysis and redesign**

The prototype system that has been developed should be tested extensively by the experts who acquired the knowledge during the acquisition phase. Then during this phase the experts can determine if the conclusions reached by the expert system are in fact valid and are answers that an expert would make from the situations being tested. The prototype system should be first tested with a couple of examples from start to finish. These examples should be of a magnitude that they determine the weaknesses of the system, knowledge base, and inference structure. The elements that normally cause poor performance are input/output characteristics, control strategies, inference rules, and the test examples used on the system. At this point any changes can be recommended and redesign of the system may be in order. Through this analysis and redesign phase, areas may be discovered by the experts where there needs to be further explanation given for the actions taken by the system in reaching a conclusion. The search paths used and reported by the expert system should be checked to ensure that they are reasonable, valid, and the correct path to take. Through this analysis process, any changes that need to be made by the expert system can be identified. Once analysis of the system has been completed, the redesign of the system can begin, and the designers can continue to build and improve the system prototype based upon the findings made during the analysis.

7. **Final analysis and fine tuning**

After the initial analysis has been accomplished and the recommended changes have been incorporated into the initial design, the system should be ready for final testing. The individuals who tested the initial system should also test the revised system.

8. **System Maintenance**

Expert systems are often used for tasks in which the input information may have to be changed, updated, or revised. Whenever this occurs, the system may need to be analyzed and tested to see if the conclusions generated by the expert system are still valid and best for the situation. The updating and additional system development should be accomplished by those persons who did the knowledge acquisition, in order to maintain system organization and systematic programming.

Attempting to replicate expert decision-making with a computer requires a well and carefully planned yet flexible design strategy. Construction of an expert system from the initial design to a working system consists of following the above loosely defined steps. Because the programming task of an expert system often includes a complex interrelationship between elements of the system, the design of an expert system may require many reworkings before an accepted performance level is achieved.

H. SUMMARY

Expert systems investigate methods and techniques for the construction of man-machine systems that have a specialized decision-making expertise. Expertise consists of knowledge about a particular domain, understanding of a domain problem, and skills at solving the problem. The formulation of knowledge in an expert system is accomplished by the knowledge engineer who gathers knowledge from public and private sources on a particular subject. This knowledge is then used to develop the basic concepts to be used in an expert system. Identification of relationships that exist between the basic concepts leads to the formulation of a rule base. The construction of an expert system prototype then enables the expert to evaluate rule base design and system execution. Final development of a full-scale system requires extensive analysis, redesign and testing.

III. DEVELOPMENT OF 'EXPERTAIR'

A. OVERVIEW

The primary goal of this thesis was to investigate the feasibility of applying an expert systems approach to non-repetitive problems in a stochastic environment. It was hoped that a set of rules could be established for a deterministic sequence of events and later adapted to the non-repetitive problem. A determination was made to utilize a military problem as a testbed for an expert system construction and evaluation. The air defense weapons employment process was to be modelled within specified parameters and subsequently evaluated using an expert system prototype. The choice of developing a model supporting military decision-making in an air defense milieu was based upon the author's combined background in tactical computers and operational experience in the air defense field. It was hoped that through the development of the model, a better understanding could be obtained on the complexities and requirements of the air defense/battle management process. In addition, it was desired that a prototype expert system could be developed for use as a framework for training devices in the air defense field.

The methodology that was used to accomplish the research paralleled the first six steps for construction of an expert system outlined in Chapter II. During the initial problem analysis, the weapons employment process was found to be a candidate problem for expert systems technology. The knowledge acquisition phase produced an initial bounding of the environment, identification of resources, and anticipated goals for tactical system employment. The implementation stage consisted of brainstorming various concepts to represent the pertinent information. These concepts were then organized into a hierarchical structure in order to identify conceptual information flow. In the initial system design, a model framework was developed to specify the direct and indirect relationships of all key concepts. An expert system development package was then utilized to organize the model framework into the working prototype 'EXPERTAIR'. The final stage of the system development included the construction of formatted system data files which were used to evaluate the operability of the prototype expert system.

Construction of 'EXPERTAIR', from initial design to working prototype, consisted of extensive programming to develop the concepts and concept relationships that existed within the weapons employment process model. The design of 'EXPERTAIR' required many reworkings before operability was achieved. Although outside evaluators were not utilized to conduct a detailed analysis of 'EXPERTAIR', it was determined by the authors of this thesis that the prototype was able to infer accurate conclusions from the input data files.

B. INTRODUCTION

Active air defense may be defined as direct defensive action taken to destroy or reduce the effectiveness of an enemy air attack. Passive air defense includes measures taken to minimize the effects of hostile air action and includes cover, concealment, and dispersion. [Ref. 10: p.9.2-5] The U.S. Marine Corps air defense mission, operationally executed by a Tactical Air Operations Center (TAOC), may be categorized into three specific tasks: area surveillance, enroute traffic control, and weapon systems employment. The task of area surveillance includes the detection, identification, and classification of acquired targets within an area of responsibility. Surveillance is primarily accomplished utilizing the TAOC's internal sensors capabilities but is routinely supplemented by external sources via data link. A summary display of the air situation is maintained and appropriate information is disseminated to other agencies. Enroute traffic control is also normally an internal function of the TAOC consisting of flight advisory, direction, monitoring, and control of itinerant aircraft. Flight plans are received and correlated with radar contacts for positive identification. Positive control is provided to launching and recovering aircraft. The third task of the air defense mission requires a direct interface with external agencies. The employment of air defense assets in the defense of a designated vital area is the intrinsic mission of a TAOC. This task integrates with the other functions of the TAOC in order to appraise the danger of an airborne threat and determine the appropriate action required to neutralize it.

There exists a multitude of weapon platforms in the U.S. military arsenal that are capable of supporting the air defense mission. Currently, that Marine Corps mission is sustained by (1) the F/A-18 and F-4 fighter aircraft, long range airborne weapon systems, (2) the IHAWK surface-to-air missile system, an intermediate range ground platform, and (3) the redeye/stinger surface-to-air missile system, a hand-held, short range weapon. The integration of these assets into a coordinated air defense system requires an extensive understanding of the characteristics of the weapon platforms and a knowledge of their tactical employment.

The decision to direct a particular weapon system against a potential threat is made by the Senior Weapons Director within the TAOC. This decision-making process requires the consideration of vast amounts of data in order to ensure a proper response. The current environment in which air defense assets may be employed has reduced the amount of time available in which to make a decision.

C. PROBLEM STATEMENT

The present requirement to analyze input information and formulate an expedient and appropriate decision under adverse conditions has severely challenged the air defense community. Current algorithmic computer devices that support the weapons employment process are inflexible and provide only limited data analysis. Total automation of the air defense mission might be a distant solution, however, a man-in-the-loop, real-time decision *aid* may provide near-term assistance. In an attempt to apply expert systems technology to the air defense problem, this chapter seeks to answer the following questions:

1. Can the air defense weapons employment process be modelled?
2. Can an expert system prototype be developed to aid and support the air defense decision-making process?

Without an ability to rapidly process large amounts of information and examine all available options, the decision-maker may be forced to make less than optimum decisions during time of war.

D. MODEL

The air defense weapons employment process entails a series of data input, evaluations, actions, and changes that are designed to bring about a desired result. Since the weapons employment process is utilized in a stochastic environment, a multitude of variables are present making it difficult to assess relationships that exist between the data input and the resultant actions. In order to define the weapons employment process for subsequent analysis, a unique representation or model of the process will be developed using specified parameters. Development of this model will allow for the establishment of a set of rules which provide deterministic conclusions to a given problem.

E. ENVIRONMENT

In order to model the air defense weapons employment process, a thorough analysis had to be made on the environment in which the process would be utilized. A description of all the parameters which may affect the process would be almost impossible, however, an initial list was defined which could be later modified if model expansion was desired. This bounding of the environment was accomplished for our model by determining a mission statement and identifying the air defense assets available to the TAOC. For our model, the TAOC was tasked to provide a limited point air defense capability for a designated vital area against a small to moderate projected threat. A secondary mission included the protection of all air defense resources. The assets available included a limited squadron of fighter aircraft (total of nine), an automated TAOC with multiple sensors, an IHAWK battery with 12 available missiles, and a platoon of redeye/stinger personnel with 20 available missiles.

The stipulation of a maximum number of available assets (aircraft/missiles) was important in the model development. It was felt that this real world limitation directly impacts the decision-making process. In order to calculate the best response to an enemy's action, the Senior Weapons Director in the TAOC must identify the operational availability of all weapon systems, determine the options available to neutralize the threat, and project future demands on assets. For our model, the final action taken to negate a threat would be partially contingent upon limited assets without anticipated resupply.

Intelligence on the enemy is another real world facet of the decision-making process. Accurate and timely information on the enemy, including type, size, and intentions, increases the probability of predicting the best response to an enemy action. For our model, generic parameters such as fighter aircraft speed and HAWK missile maximum kill ranges were utilized in the specification of enemy and friendly capabilities in order to allow the thesis classification to remain UNCLASSIFIED. However, in order to fully implement a tactical air defense decision aid, it would be necessary to develop a knowledge base founded upon actual systems parameters.

Information contained in an air defense mission directive usually includes a specified surveillance sector or Area of Responsibility (AOR). The determination of a TAOC's AOR, whether 40° or a full 360° sector, will assist in the development of a sensor emplacement and tactical utilization plan. An objective of this plan is to ensure maximum area coverage while minimizing transmissions in the electromagnetic spectrum by radiating only in the required sector. For our model, an AOR subtending the arch from 000°, or North, to 140° or Southeast was directed.

F. WEAPON SYSTEMS EMPLOYMENT

After the establishment of the AOR, the next task to be accomplished in the model development was the determination of the optimum location for weapon systems employment. Important factors in this determination included weapon systems capabilities, support limitations, surveillance and communications restrictions, and enemy characteristics. These variable factors affect the weapons employment process in an operational TAOC and should be considered. For our model, all air defense assets were located at the center of the vital area (area to be protected).

In the development of an air defense plan, the Senior Weapons Director analyzes available weapon systems capabilities to help determine the boundaries of a designated destruction area. The destruction area is an area in which the enemy will be destroyed or defeated. It is divided into a missile intercept zone, cross-over zone, and air intercept zone. The destruction area begins at the edge of the vital area. Its size varies with the capabilities of the air defense system, the engagement sequence, and the surveillance capabilities. The missile intercept zone (MIZ) identifies the area where tracking solutions and engagement sequences from surface-to-air missile units will occur. This zone may be determined by ground missile system maximum kill ranges, terrain limitations, and weather. The crossover zone is a buffer area and identifies a region where airborne platforms should start to discontinue target intercepts and allow the ground-based systems to begin target tracking. The region beyond the crossover zone and bounded by the AOR is designated as the air intercept zone (AIZ) and constitutes the area of airborne target engagement.

A determination must be made on the location, number, and composition of fighter holding positions or Combat Air Patrol (CAP) stations that are required to protect the vital area. The kill probability required for air defense has the greatest influence on CAP employment. Other real world factors which influence the stationing of CAP's are: [Ref. 11: p. 161]

1. The required probability of detection.
2. The capability of surface radar to achieve this probability of detection.
3. The effective kill probabilities of secondary, defensive surface-to-air missile systems.
4. Size and orientation of AOR.
5. The number of fighter aircraft available.

Examination of these factors led to the determination of the model's optimum CAP station locations. This data, along with AOR specifications, asset locations, and designation of the destruction area, has been depicted on a tactical working area map in Figure 3.1. This map can be utilized when briefing the environment of the weapon systems employment process. The locations of the CAP stations are denoted by a bearing and range from system center (the TAOC's location).

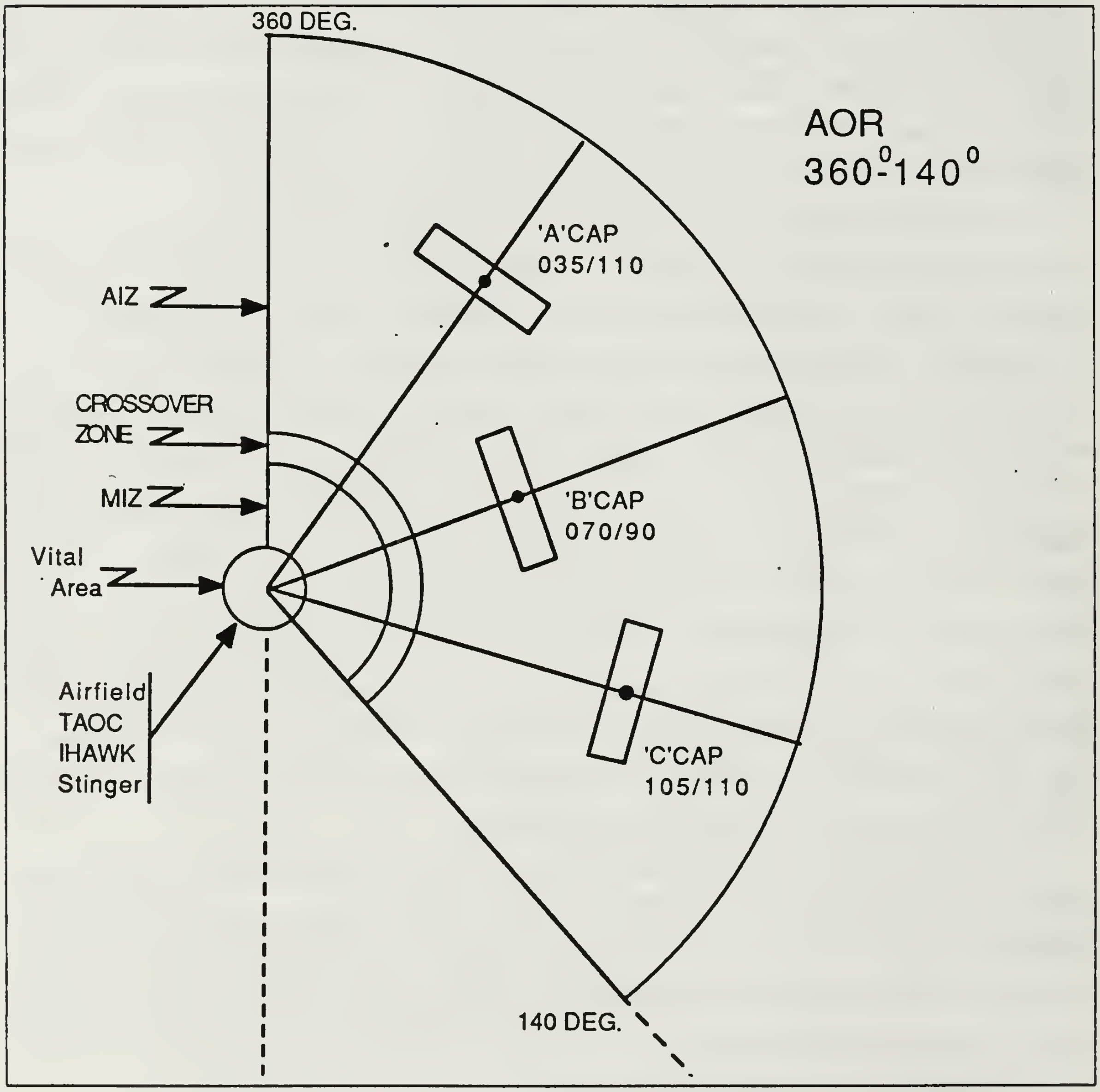

Figure 3.1 Tactical Working Area.

The development of the tactical working area map completed the analysis and identification of the environment. The air defense role and the system requirements were defined and bounded. The next step used in the knowledge acquisition process dealt with the identification and formalization of concepts used to represent the given knowledge.

G. KNOWLEDGE CONCEPTS

In order to model a large-scale system such as the air defense weapons employment process, it was necessary to define the separate components that aggregate into the system as a whole. Decomposition of the system into individual elements allowed for a more precise definition of the specific system structure. In the development of the model, the weapons employment process was envisioned to consist of six abstract ideas or concepts. These interactive concepts were designated system status information, target, fighter, airborne, and probability of intercept information, and specific actions available to neutralize a threat. The concepts were then further broken down into properties and roles in an attempt to relate them to specific data items that currently existed or could be obtained through present air defense mechanisms. A property is a quality of a concept while a role is a component of a concept. A role usually specifies a value or a number restriction or constraint.

The *system status* concept included information on the current state of alert, the weapons condition as basis for rules of engagement, and the status of the individual air defense assets. The limitations on available assets were defined here. This information could be compiled utilizing the internal TAOC command structure along with communication links to the external agencies.

The concepts of *target* and *fighter* contained data developed within the TAOC. Most of this information could be automatically obtained or operator entered utilizing present TAOC equipment. The target roles included identity, location, speed and movement of target aircraft while the fighter roles identify status, location, and targeting information of fighters. Both concepts only allowed input from two separate aircraft or groups of aircraft, thus leading to identical restrictions on targets #1 & #2 and fighters #1 & #2. A maximum of two fighter sections versus two target groups could be analyzed during a single data input. These limiting factors were introduced to restrict the number of rules required to substantiate the model. However, the expansion of this concept could be easily achieved to support larger numbers of aircraft by using similar rules applicable to the single aircraft.

The *airborne* concept described the number of targets and fighters that were airborne at data input. This concept limited duplications of information queries within the rule file by condensing the number of data items applicable to a rule.

The *probability of intercept* (POI) concept consisted of applicable system status, target, and fighter data input, and described an intercept probability for the airborne fighters. This concept was also limited to information on a maximum of two fighter and target groups. The concept parameters described the intercept probabilities for fighter #1 versus target #1, fighter #2 versus target #1, etc. The probability categories ranged from zero to low, medium, and high POI's. These categories corresponded to a fighter's ability to cover or engage a specified target. A high POI corresponded to a 80-100% probability of intercepting a target and maneuvering to a position from which to commit on a target. A medium POI : 50-79%; low POI : 1-49%; zero POI : 0%..

The final knowledge concept of *neutralizing actions* consisted of the various actions that could be taken to neutralize a potential threat. These actions paralleled the decisions that a Senior Weapons Director would be required to make in a tactical situation. The final actions ranged from continuing normal surveillance to engagement of target #2 with fighter #1 while fighter #2 engages with target #1 or some variation.

Recognition of the relationships that exist between the concepts and determination of the data structure completed the modelling of the weapons employment process. The model shown in Figure 3.2 represents an input of data, obtained from a specified air defense environment, that is distributed to applicable concepts. The final concept, which directs an appropriate weapons employment action, is an input to the entire process and indicates continuous flow of information.

H. THE 'EXPERTAIR' PROTOTYPE

Having a structured understanding of the air defense weapons employment process, an attempt was made to apply an expert systems approach toward the development of a decision support aid for that military problem. A prototype, designated 'EXPERTAIR', was developed to simulate the functionings of an expert system operating in a tactical environment utilizing existing Marine Corps air defense equipment for data input. The prototype was evaluated for operability and was found to be simple to operate, user friendly, and provide concise expert conclusions based upon the current situation. A detailed analysis of 'EXPERTAIR' was beyond the scope of this thesis, however, the conclusions produced by the expert system prototype were identified by the authors as accurate and appropriate.

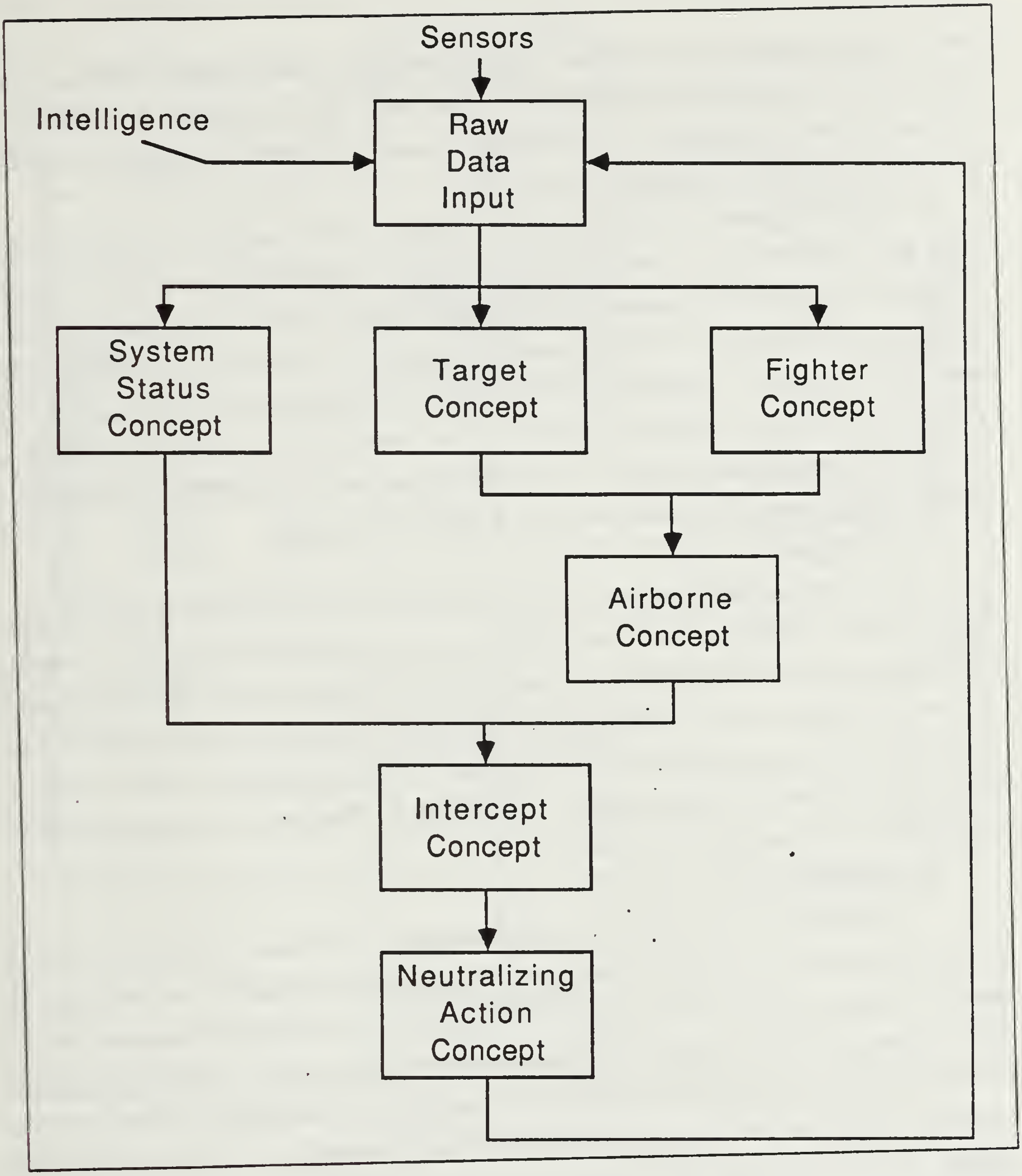

Figure 3.2 Conceptual Model of Weapons Employment Process.

During the initial design of 'EXPERTAIR', the concepts that were identified in the weapons employment process model were configured into an expert system format using the *Arity/Expert Development Shell* as a development tool. The *Arity* shell met the desired implementation criteria of uncomplicated, user friendly software coding format and instructions. In addition, it provided a large capability to evaluate and expand system performance. *Arity* software may be executed utilizing a standard personal computer with 610K of memory.

The development of 'EXPERTAIR' entailed the construction of various software files that conformed to a structured *Arity* format. Implementation of the completed 'EXPERTAIR' prototype required the execution of the *Arity/Expert Development Shell* software in conjunction with the formatted files. The structure of 'EXPERTAIR', depicted in Figure 3.3, displays the different files that formed the final prototype. The following discussion reviews the content of the files while the actual file listings are found in Appendixes A through F. A software dictionary is presented in Appendix G which defines some of the terms used the 'EXPERTAIR' files.

1. Front-End File

A front-end file contains the code that controls the execution of the developed expert system. The contents of the 'EXPERTAIR' front-end file (Appendix A), written in Prolog, specify the procedures for execution of the database files and the format of the user's terminal screen display. In addition, the execution of the 'EXPERTAIR' conversion program (Appendix B) is controlled by the front-end file. Also written in prolog, the conversion program converts a DBASE III Plus file into a Prolog file which can then be evaluated by the system.

2. Taxonomy File

A taxonomy is a structured representation of the general information used in an expert system. The basic purpose of the taxonomy language is for defining the concepts that are used in a developed expert system. The taxonomy language defines these concepts in a form that can be interpreted by the computer. The strategy used for constructing a taxonomy file involves the mapping of relationships of the identified concepts. Every element that is involved in the decision process must be identified and defined in the taxonomy file. From this point, elements to be described in the taxonomy need to be described in terms of taxonomy definitions.

A taxonomy file has within it two distinct sets of information: type declarations and concept definitions. Type declarations are declarations of all properties and roles used in a taxonomy. Concept definitions contain the identification and description of the concepts and properties or roles they may have. The amount of information conveyed in a concept definition may have a wide variance. All concept

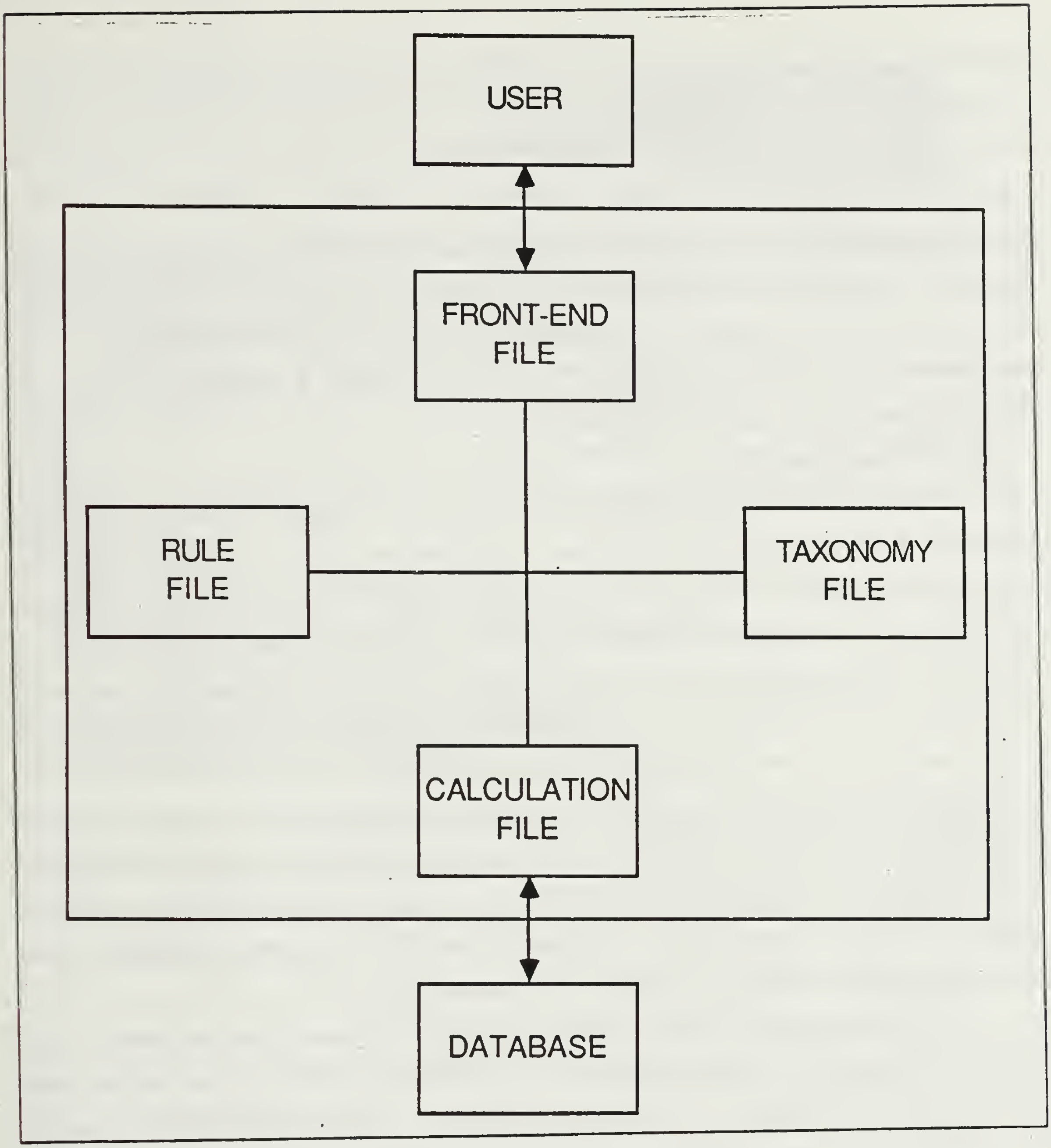

Figure 3.3 EXPERTAIR Structure.

definitions are based on a simple taxonomy language format, although they may also be quite complex.

An example concept definition from the 'EXPERTAIR' taxonomy file is shown below:

```
define primitive status with
    defcon = [white,yellow,red] and
    wepcon = [tight,free] and
```

```
runw_stat = [op,nonop] and
acft_avail = (0,9) and
taoc_stat = [op,deg,nonop] and
hawk_stat = [op,deg,nonop,assign] and
hawk_msl = (0,15) and
sting_msl = (0,24).
```

This example depicts the description of the *system status* concept using the taxonomy language. The roles of the concept (defcon, wepcon, runw_stat, etc.) are listed along with the value and number restrictions of each of the roles. Definitions for the concepts and roles utilized in 'EXPERTAIR' can be found in Appendix G while the entire taxonomy file is listed in Appendix C.

3. The Rule Base File

The rule base file consists of a number of control options, rules and synonyms. Rules describe the interaction of the concepts defined in the taxonomy file. Through the evaluation of these rules, the expert system is able to conclude a solution based upon the problem put forth. A rule is composed of an antecedent and a consequent which establish goals between concepts. A goal is a condition that can be proved either true or not true. The consequent of a rule is determined to be true if each of the goals in the antecedent are determined by the expert system to be true. Consequently, if any of the goals in the antecedent cannot be shown to be true then the rule will fail. The consequent of a rule will always be a single goal and will always be listed before the antecedent. The antecedent of a rule can be one or more goals. A keyword *if* is used to separate the consequent from the antecedent, and the keyword *and* is used to separate each of the goals in the antecedent.

Synonyms defined in the rule base file must be different from the name of any other concept described in the taxonomy file. Albeit, the synonym may be the same name as the rule it describes. The synonym definitions must appear before the use of the synonym in a rule. Control options used in the rule base file allow for the modification of the features used in the expert system in order to customize the way the system runs.

Specific considerations that affect a rule base development include:

- Analysis of rule base order
- Relationship of rule base size to execution time
- Consequence of multiple goals
- Consequence of omitted rules
- Consequence of insufficient data input

The order in which the rules are specified and the size of the rule base may make a significant difference in total system execution time (time to reach a decision). In addition, the omission of rules or addition of multiple goals to satisfy a single rule may affect the versatility and efficiency of an expert system. The problem of insufficient or incorrect data input should also be taken into account. In an environment in which time is of critical importance, an attempt should be made to optimize the rule base structure through detailed testing and evaluation.

The objective of 'EXPERTAIR's rule base development was to identify the interactions between the concepts listed in the taxonomy file. The rule base was not envisioned to be exhaustive but rather thorough in covering a wide range of conceivable air defense situations. An example from the 'EXPERTAIR' rule base is shown below:

```
the neutr_action of airdefense is ewh1
        if
the hawk_stat of system_status_info is [op,deg] and
the hawk_msl of system_status_info is X and
        X > 0 and
the tar1_class of target_info is hostile and
the airborne_tar of airborne_info is one and
the airborne_flt of airborne_info is one and
the prob11_intercept of intercept_info is zero.
```

In this example, the consequent of the rule is the decision to neutralize a single target (threat) with a HAWK missile. The antecedent of the rule is made up of the following conditions: (1) The HAWK missile system must be operational or degraded, (2) At least one HAWK missile must be available, (3) The target must be classified HOSTILE, (4) Only one target is airborne, (5) Only one fighter is airborne, and (6) The probability of the airborne fighter intercepting (and destroying) the target is zero. If all six conditions of this rule are found to be true, then the consequent of the rule is also true. A complete listing of 'EXPERTAIR's rule base is given in Appendix D.

4. Calculation File

The calculation file determines the value of a goal which has been presented in the rule base file. Although not required for expert system execution, the calculation file is utilized to reduce execution time by continuously calculating and updating goal values. If the calculation file were not used, the goal values for each rule in the rule base would have to be individually calculated, usually causing duplication of effort.

Using the rule base example cited above, the 'EXPERTAIR' calculation file continuously identifies, calculates, and updates at data input all values for expert system evaluation to include the HAWK missile system status, the number of missiles available, the classification of the airborne target, etc. These values are then available for all rules identified in the rule base. The calculation file that was developed for 'EXPERTAIR' is found in Appendix E.

5. Database File

Database files were constructed for 'EXPERTAIR' as a means of simulating data input to our prototype. Written in Prolog, these files, excluding Database XPT1 (file #1), were internal to the 'EXPERTAIR' prototype and were utilized as a means of testing system operability. The files are representative of actual data that could be derived in an air defense conflict. The majority of the data necessary to provide input to our prototype is presently generated in a TAOC, however, the hardware that would be required to interface with a decision support expert system does not exist.

The second objective of our thesis was to examine the feasibility of interfacing an expert system prototype with an outside system. Accomplishment of this objective would give more credibility to the near-term implementation of an expert system. 'EXPERTAIR' was developed with a capability to accept data input from outside the existing prototype, evaluate the data, and provide an expert decision. This function was tested using Database XPT1, a database file written in DBASE III Plus, and vis-a-vis the conversion file, it was translated into a prolog file for use by our prototype. All database files used in testing 'EXPERTAIR' are described in the next chapter while the actual file listings are given in Appendix F.

I. SUMMARY

The application of an expert systems approach to non-repetitive problems in a stochastic environment may have relevance to certain military tasks. The establishment of a set of rules based upon a deterministic sequence of events may provide a means for solutions to a non-repetitive problem. The modelling of the air defense weapons employment process provided a framework for the development of an expert system prototype 'EXPERTAIR'. The model development was achieved through a bounding of the operating environment and identification of concepts used to represent the weapons employment process. Concept relationships were then identified and expressed in a rule base file. 'EXPERTAIR' is comprised of software files, including the rule base file, which dictate date interpretation, expert system execution, and the user's screen presentation. The files were constructed in a specific format using an expert system development package. The prototype was evaluated for operability using sample data files for input and was found to produce accurate conclusions based on a given scenario.

IV. EXECUTION OF 'EXPERTAIR'

A. OVERVIEW

An evaluation of the system operability of 'EXPERTAIR' was accomplished utilizing a sample scenario. Examples representative of an actual air defense environment were constructed in the form of database files. These files provided a comprehensive array of situations, within the limitations of the weapons employment process model, in which an air defense decision support system might be tasked to operate. Consideration was given to the environment in which our prototype may be utilized. A TAOC is updated with tactical information every six to ten seconds depending upon the rotation speed of the sensors (radars). Additionally, other information is made available through intelligence networks and communications with other agencies. An expert system, ideally, would be able to accept, process, and provide a recommendation (decision) for each new data update. A stagnant or low intensity conflict in which data remained constant may result in repetitious recommendations such as 'do nothing' or 'continue normal surveillance activities'. However, a high intensity conflict would require an analysis of every accessible batch of data input in order to keep up with a rapidly changing environment. A critical decision to be made should be based upon the most current information available to a system.

The execution of 'EXPERTAIR' consisted of evaluating the fifteen database files outlined in Appendix F. These files consisted of individual snap-shot situations along with sequences of events (3-4 files). Each file was processed, developing individual conclusions to the data input. The current data input was initially displayed on a monitor screen while the information was being processed. At the completion of the data processing, the expert system conclusions were displayed under the screen heading 'Recommendation'.

The remainder of this chapter depicts the results of the execution of 'EXPERTAIR' to include data input, screen display, and recommendations for the fifteen files. Each file is introduced with a background on the current tactical situation prior to making a decision. One capability that is characteristic of expert systems is the ability to obtain explanations on all intermediate and final decisions. An example of this capability is given at the conclusion of this chapter.

B. EXECUTION LISTINGS

File Number (1): This file was constructed utilizing DBASE III Plus and converted into a useable system format. It represents a low threat environment. Situation is calm. TAOC has identified no targets in the AOR. All fighters are on the runway. All assets are fully operational.

CURRENT AAW STATUS

Defense Condition:	White	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	0	Track Number (2):	0
Bearing:	0	Bearing:	0
Range:	0	Range:	0
Size:	0	Size:	0
Speed:	0	Speed:	0
Class:	Unknown	Class:	Unknown
Movement:	Unknown	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	Onrun	Location:	Onrun
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Continue normal surveillance; no immediate threat is present; no decision is required.

File Number (2): New file. Also a low threat environment. The TAOC has identified a target in the area and has determined it to be a probable (assumed) friend.

CURRENT AAW STATUS

Defense Condition:	White	Weapon Condition:	Tight
Runway Status:	Op	Avail Fighter AC:	9
TAOC Status:	Op	Hawk Status:	Op
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4001	Track Number (2):	0
Bearing:	135	Bearing:	0
Range:	200	Range:	0
Size:	1	Size:	0
Speed:	350	Speed:	0
Class:	Asfriend	Class:	Unknown
Movement:	Closing	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	Onrun	Location:	Onrun
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Continue normal surveillance; no immediate threat is present; no decision is required.

File Number (3): New file. A single target has been detected in the area. The TAOC has classified the target as a probable (assumed) enemy. There are no other aircraft airborne.

CURRENT AAW STATUS

Defense Condition:	White	Weapon Condition:	Tight
Runway Status:	Op	Avail Fighter AC:	9
TAOC Status:	Op	Hawk Status:	Op
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4002	Track Number (2):	0
Bearing:	110	Bearing:	0
Range:	185	Range:	0
Size:	1	Size:	0
Speed:	375	Speed:	0
Class:	Asenemy	Class:	Unknown
Movement:	Orbiting	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (1):	Fullop
Flt Size:	1	Flt Size:	1
Location:	Onrun	Location:	Onrun
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Launch Fighter #1 and assign to Cap C. Attempt to determine intentions of target. If target is identified as a threat, set Defense Condition to yellow.

File Number (4): Continuation file. Defense condition has been increased to yellow. A single fighter has been launched and is holding at CAP 'C'. The TAOC has identified and classified a second section of targets in the area. Both targets are not considered high priority targets due to their movement (orbiting).

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4002	Track Number (2):	4003
Bearing:	110	Bearing:	028
Range:	185	Range:	170
Size:	1	Size:	2
Speed:	375	Speed:	400
Class:	Asenemy	Class:	Asenemy
Movement:	Orbiting	Movement:	Orbiting

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	CCap	Location:	Onrun
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Continue surveillance of target/s. Launch (as required) to position fighters at CAPS's A & B. If a hostile act is commited, then engage target.

File Number (5): Continuation file. Single fighters are located at CAP's 'A' and 'B'. The single target has become a greater threat due to decreasing range to the vital area. No hostile act has yet been committed.

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4002	Track Number (2):	4003
Bearing:	110	Bearing:	030
Range:	170	Range:	190
Size:	1	Size:	2
Speed:	425	Speed:	400
Class:	Asenemy	Class:	Asenemy
Movement:	Inbound	Movement:	Outbound

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	ACAP	Location:	BCAP
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Cover threat (target 1) with fighter #2 and attempt to determine intentions. If a hostile act is committed or if target enters the restricted area, then engage target. Inform HAWK and Stinger units of situation.

File Number (6): Continuation file. The TAOC had assigned the closest asset (fighter #2) to cover the inbound threat. However, the target is now indicating an outbound movement (simulating a feint). No other targets in the area.

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4002	Track Number (2):	0
Bearing:	110	Bearing:	0
Range:	175	Range:	0
Size:	1	Size:	0
Speed:	425	Speed:	0
Class:	Asenemy	Class:	Unknown
Movement:	Outbound	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	ACAP	Location:	Assign
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	65
Assigned to TN:	0	Assigned to TN:	4002

RECOMMENDATION: Cancel cover/engage assignment. Target is no longer an immediate threat. Return fighters to CAP or return to base and replace as required.

File Number (7): New file. Single fighters located on CAPs. Two different targets have been detected by the TAOC. Both are possible threats due to movement. All assets are fully operational. A sample presentation of the the explanation facility is presented at the conclusion of this file.

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4010	Track Number (2):	4012
Bearing:	050	Bearing:	130
Range:	190	Range:	175
Size:	1	Size:	1
Speed:	400	Speed:	400
Class:	Asenemy	Class:	Asenemy
Movement:	Inbound	Movement:	Inbound

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	ACAP	Location:	BCAP
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Cover threat (target 1) with fighter #1 and (target 2) with fighter #2 and attempt to determine intentions. If a hostile act is committed or if target enters the restricted area, then engage target. Inform Hawk and Stinger units of situation.

File Number (8): Continuation file. The TAOC has the two fighter sections assigned to cover the inbound targets. Target movement remains the same.

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4010	Track Number (2):	4012
Bearing:	050	Bearing:	128
Range:	180	Range:	165
Size:	1	Size:	1
Speed:	400	Speed:	410
Class:	Asenemy	Class:	Asenemy
Movement:	Inbound	Movement:	Inbound

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	Assign	Location:	Assign
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	78	Range to Tar_TN:	57
Assigned to TN:	4010	Assigned to TN:	4012

RECOMMENDATION: Continue to monitor target assignment/s. If a hostile act is committed, then engage target/s. Update HAWK and Stinger units.

File Number (9): New file. Two sections of fighters are on CAP. The TAOC detects a fast low flying target moving toward the vital area. Target has been classified hostile but is outside the range of CAP aircraft.

CURRENT AAW STATUS

Defense Condition:	Yellow	Weapon Condition:	Tight
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4020	Track Number (2):	0
Bearing:	130	Bearing:	0
Range:	60	Range:	0
Size:	1	Size:	0
Speed:	600	Speed:	0
Class:	Hostile	Class:	Unknown
Movement:	Inbound	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	2	Flt Size:	2
Location:	ACAP	Location:	BCAP
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Attempt to engage threat (target 1) with HAWK missile when target is in range. Inform Stinger units of the situation.

File Number (10): Continuation file. Defense and weapon condition have upgraded. HAWK has been assigned to engage target. Target is still outside of HAWK's parameters. No other targets in the area.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	Assign
Hawk Missile Inv:	12	Stinger Msl Inventory:	20

TARGET INFORMATION

Track Number (1):	4020	Track Number (2):	0
Bearing:	130	Bearing:	0
Range:	40	Range:	0
Size:	1	Size:	0
Speed:	600	Speed:	0
Class:	Hostile	Class:	Unknown
Movement:	Inbound	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	2	Flt Size:	2
Location:	ACAP	Location:	BCAP
Avail Msls:	6	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Continue to monitor HAWK target engagement. Update Stinger units on situation. Be prepared to flush airfield.

File Number (11): New file. High threat environment. Air defense missile inventory has decreased. The TAOC detects a section of low flying targets classified hostile. All assets are operational.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	10	Stinger Msl Inventory:	18

TARGET INFORMATION

Track Number (1):	4024	Track Number (2):	0
Bearing:	025	Bearing:	0
Range:	135	Range:	0
Size:	2	Size:	0
Speed:	450	Speed:	0
Class:	Hostile	Class:	Unknown
Movement:	Unknown	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	DEG	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	ACAP	Location:	BCAP
Avail Msls:	4	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Attempt to engage threat (target 1) with fighter #1. If no kill prior to entrance of MEZ, break off assignment and engage with HAWK. Inform Stinger units of situation.

File Number (12): New file. High threat environment. The TAOC has detected two sections of target aircraft. A section of fighter aircraft is located at CAP 'A' while a single is at CAP 'B'.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	10	Stinger Msl Inventory:	18

TARGET INFORMATION

Track Number (1):	4027	Track Number (2):	4030
Bearing:	75	Bearing:	60
Range:	160	Range:	170
Size:	4	Size:	2
Speed:	480	Speed:	500
Class:	Asenemy	Class:	Hostile
Movement:	Inbound	Movement:	Inbound

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	DEG
Flt Size:	2	Flt Size:	1
Location:	ACAP	Location:	BCAP
Avail Msls:	4	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Attempt to engage threat (target 2) with fighter #1 and (target 1) with fighter #2. If no kill prior to entrance of MEZ, break off assignment and engage with HAWK. Inform Stinger units of situation.

File Number (13): Continuation file. The TAOC has assigned the fighters to engage the hostile targets. The target movement continues to be inbound. The targets are still outside fighter weapon parameters.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	OP	Avail Fighter AC:	9
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	10	Stinger Msl Inventory:	18

TARGET INFORMATION

Track Number (1):	4027	Track Number (2):	4030
Bearing:	75	Bearing:	62
Range:	130	Range:	140
Size:	4	Size:	2
Speed:	480	Speed:	500
Class:	Hostile	Class:	Hostile
Movement:	Inbound	Movement:	Inbound

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	DEG
Flt Size:	2	Flt Size:	2
Location:	Assign	Location:	Assign
Avail Msls:	4	Avail Msls:	6
Range to Tar_TN:	25	Range to Tar_TN:	15
Assigned to TN:	4030	Assigned to TN:	4027

RECOMMENDATION: Continue to monitor target engagement/s. If no kill prior to entrance of MEZ, break off with flight assignment and engage with HAWK. Update Stinger units.

File Number (14): Continuation file. Fighter #1 has expended all missiles while fighter #2 has been shot down. Four targets have been destroyed while two continue inbound.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	OP	Avail Fighter AC:	6
TAOC Status:	OP	Hawk Status:	OP
Hawk Missile Inv:	10	Stinger Msl Inventory:	18

TARGET INFORMATION

Track Number (1):	4027	Track Number (2):	0
Bearing:	75	Bearing:	0
Range:	100	Range:	0
Size:	2	Size:	0
Speed:	500	Speed:	0
Class:	Hostile	Class:	Unknown
Movement:	Inbound	Movement:	Unknown

FIGHTER INFORMATION

Status (1):	Fullop	Status (2):	Fullop
Flt Size:	1	Flt Size:	1
Location:	ACAP	Location:	ONRUN
Avail Msls:	0	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Attempt to engage threat (target 1) with HAWK missile when target is in range. Inform Stinger units of situation.

File Number (15): New file. The airfield runway and the HAWK battery have been severely damaged. The TAOC, although degraded, has detected two raids of inbound hostile targets.

CURRENT AAW STATUS

Defense Condition:	Red	Weapon Condition:	Free
Runway Status:	NonOP	Avail Fighter AC:	2
TAOC Status:	DEG	Hawk Status:	NonOP
Hawk Missile Inv:	3	Stinger Msl Inventory:	6

TARGET INFORMATION

Track Number (1):	4040	Track Number (2):	4042
Bearing:	38	Bearing:	75
Range:	45	Range:	38
Size:	4	Size:	4
Speed:	600	Speed:	500
Class:	Hostile	Class:	Hostile
Movement:	Inbound	Movement:	Inbound

FIGHTER INFORMATION

Status (1):	DEG	Status (2):	Fullop
Flt Size:	1	Flt Size:	2
Location:	CCAP	Location:	Onrun
Avail Msls:	0	Avail Msls:	6
Range to Tar_TN:	0	Range to Tar_TN:	0
Assigned to TN:	0	Assigned to TN:	0

RECOMMENDATION: Attempt to engage target/s with Stinger missile units. Be prepared to flush airfield.

C. EXPLANATION FACILITIES IN 'EXPERTAIR'

'EXPERTAIR' contains an explanation facility which, on request, provides the decision-maker an opportunity to identify the data and sequence of rules utilized in deriving the final recommendations. The facility applies a hierarchical structure which allows a user to ascertain a desired level of detail of rule explanation. By continuing to request the facility to explain the given output, the user would ultimately obtain a list of all rules employed in the development of the decision. The following example of the explanation capability lists the derivation of the final decision to database file #7.

EXPLAIN

[How was neutr_action of air_defense-7 judged to be Cover threat (target 1) with fighter #1 and (target 2) with fighter #2 and attempt to determine intentions. If a hostile act is committed or if target enters the restricted area, then engage target. Inform HAWK and Stinger units of situation. (1.000) ?]

Since [1] the wepcon of status-7
is tight
and [2] the tar1_bear of target-7
is 50
and 50 < 70
and [3] the tar1_class of target-7
is asenemy or unknown
and [4] the tar2_bear of target-7
is 130
and [5] the tar2_class of target-7
is asenemy or unknown
and [6] the prob11_intercept of intercept-2
is med or high
and [7] the prob22_intercept of intercept-2
is med or high

then the neutr_action of the airdefense-7 is Cover threat (target 1) with fighter #1 and (target 2) with fighter #2 and attempt to determine intentions. If a hostile act is committed or if target enters the restricted area, then engage target. Inform HAWK and Stinger units of situation.

EXPLAIN **(number of value) ? 1**

[How was the wepcon of status-7 judged to be tight?]

• This value was determined by your response to a question (in the form of a data input).

EXPLAIN (**number of value**) **? 2**

[How was tar1_bear of target-7 judged to be 50?]

• This value was determined by your response to a question (in the form of a data input).

EXPLAIN (**number of value**) **? 3**

[How was tar1_class of target-7 judged to be asenemy or unknown?]

• This value was determined by your response to a question (in the form of a data input).

EXPLAIN (**number of value**) **? 4**

[How was tar2_bear of target-7 judged to be 130?]

• This value was determined by your response to a question (in the form of a data input).

EXPLAIN (**number of value**) **? 5**

[How was tar2_class of target-7 judged to be asenemy or unknown?]

• This value was determined by your response to a question (in the form of a data input).

EXPLAIN (**number of value**) **? 6**

The values of prob11_intercept of intercept-2 are :

1) med / -1.0
2) high / 1.0

EXPLAIN (**number of value**) **? 2**

(for 'high')

[how was prob11_intercept of intercept-2 judged to be high?]

```
Since [1] the tar1_bear of target-7
                 is 50
  and 50 < 70
  and [2] the tar1_range of target-7
                 is 190
  and 190 > 119
  and [3] the tar1_move of target-7
                 is inbound or closing
  and [4] the fit1_stat of fighter
```

is fullop

and [5]the fit1_loc of fighter

is onrun or en_cap or acap

then the prob11_intercept of the intercept-2 is high

EXPLAIN (number of value) ? 7

The values of prob22_intercept of intercept-2 are :

1) med / -1.0

2) high / 1.0

EXPLAIN (number of value) ? 2

(for 'high')

[How was prob22_intercept of intercept-2 judged to be high?]

Since [1] the tar2_bear of target

is 130

and 130 > 70

and 130 < 140

and [2] the tar2_move of target-7

is 175

and 175 > 119

and [3] the tar2_move of target

is inbound or closing

and [4] the fit2_stat of fighter-3

is fullop

and [5] the fit2_loc of fighter-3

is onrun or en_cap or bcap

then the prob22_intercept of the intercept-2 is high

D. SUMMARY

The execution of 'EXPERTAIR' consisted of evaluating fifteen database files which depicted events in a simulated air defense scenario. Each file was processed by identifying the input data in terms of concepts defined in 'EXPERTAIR's taxonomy file, comparing those concepts with the rules established in the rule base file, and displaying the resultant expert system conclusions. During the processing of the files, the input data was displayed on a terminal screen. At the completion of processing, the conclusions were identified and displayed as system 'recommendations'. 'EXPERTAIR's explanation facility provides the decision-maker an opportunity to identify the data and sequence of rules utilized in deriving the final recommendations.

V. FUTURE EXTENSIONS AND CONCLUSIONS

A. OVERVIEW

During the development of 'EXPERTAIR', the multiple parameters of the environment were bounded in order to keep the scope of the scenario within the realm of this thesis. However, now that the basic prototype has been constructed, the limitations imposed by this thesis can be gradually lifted for future development. The concepts and strategy that went into the development of 'EXPERTAIR' could allow for further research in three distinct areas: (1) A larger and more extensive air defense environment, (2) Training and tutoring of air defense personnel, and (3) Further incorporation of external sources.

B. FUTURE AIR DEFENSE ENVIRONMENTS

Now that the basic expert system prototype has been developed, future research and continued maturation can expand the role, complexity, and usefulness of the weapons employment process model in the following areas:

- Expand the air defense scenario to include an electronic warfare environment and introduction of an anti-radiation missile threat
- Development of an advanced screen display updating any parametric changes in the air defense scenario
- Interface the system with atmospheric condition prediction systems
- As an aid to other air defense missions to include surveillance and itinerant air traffic control

1. Expansion of Air Defense Scenario

The air defense scenario that was used in development of the weapons employment process model was limited in scope. Follow-on research to the model could expand and increase the number of parameters interpreted and evaluated in an air defense environment. Incorporation of an electronic warfare environment and the threat of anti-radiation missiles are two areas where the model can be expanded. A broader environment and increasingly complex parameters that the model must consider may lead to the development of a full-scale expert system.

2. Advanced Screen Display

The development of 'EXPERTAIR' included a fundamental screen display for showing the results of the information that was being processed vis-a-vis database files.

A listing of the screen displays developed in 'EXPERTAIR' was presented in Chapter 4. Utilizing the prototype, every time a new scenario was evaluated, a new screen would appear with the information that was processed, as well as a recommended course of action. A follow-on study could further develop the screen management to allow for continuous updating of information in the air defense environment. This new information could be highlighted in color or graphics to indicate intensity level and the situation of the hostile environment, giving visual aids to the operators of the system.

3. Interface with Atmospheric Condition Predictions

The prototype developed only dealt with weapons employment in a limited air defense environment. Further development could take into consideration changing atmospheric conditions which have an impact on the decisions to be made regarding the environment. The expert system prototype could be further developed to incorporate these atmospheric condition predictions for evaluation before a decision is made and a recommended course of action is given.

4. Expansion of Man-Machine Interface

The man-machine interface incorporated in 'EXPERTAIR' is very limited. The interface is through operator keyboard interaction, using the explanation facility of the expert system. This area could be greatly enhanced and expanded by using some alternative type of man-machine interaction such as voice recognition. Further development in this area could allow for the expert system model to be responsive to voice, both in giving and receiving information about a changing environment.

5. Additional Mission Aids

'EXPERTAIR' could also be further designed and developed to aid other missions of an air defense unit. One area might be surveillance of an operating area where information is received via sensors and is interpreted and evaluated utilizing a rule base established on identification and classification criteria. Another mission where this prototype could be used or expanded includes the control of airborne itinerant aircraft. Data link instructions for transiting aircraft or aerial refuelers could be developed utilizing a traffic direction and control rule base.

C. TRAINING AND TUTORING

The expert system prototype which has been developed could be used with further research and development as an instructional/tutoring aid. With further expansion of the air defense parameters to simulate a more comprehensive combat environment, this prototype could help develop skills in the following areas:

- Situational analysis
- Threat recognition
- Target classification

1. Situational Analysis

'EXPERTAIR' could be expanded and developed in such a manner as to aid in the training of air defense personnel to recognize, interpret, and analyze information presented in a changing air defense environment. The student could then ascertain a course of action which would be checked against the recommendation given by the expert system.

2. Threat Recognition

Threat recognition training could be developed along the same avenue as the situational analysis objective. Students of air defense training could be taught to identify a changing air defense environment and recognize indications of threat advances/movement.

3. Target Classification

'EXPERTAIR' could also be developed to incorporate more information and rules about hostile targets. An extensive knowledge base, established on actual weapon systems platform parameters, could be developed as a viable training aid for air defense personnel allowing realistic training in the area of target analysis and classification.

This prototype used in the training arena could be useful either as an individual tutoring/training device or in combat crew training. Expansion and development of this prototype would allow for ground training without the costly use of aircraft. Presently in the U.S. Marine Corps, there is a primary level of readiness in air defense qualification, however, advanced training is not at a level that might be obtained through the use and further development of expert systems technology.

D. INCORPORATION OF EXTERNAL SOURCES

The second objective of this thesis was to examine the feasibility of interfacing an expert system prototype with an external system. Our research and testing of this question was accomplished utilizing DBASE III Plus software as an external source of data input. Using DBASE III Plus software, a file was built containing information from one of the constructed air defense scenarios. This file was then converted by 'EXPERTAIR' to draw the required inputs for processing and analysis and concluding with a recommended course of action to take. Our prototype was able to effectively use this external source which gives further credibility to continued research and system expansion. Additional external systems could also be tested to interface with 'EXPERTAIR' to include actual sensors gathering and receiving information about the environment.

E. CONCLUSIONS

The primary goal and emphasis area of this thesis was to determine the feasibility of using an expert systems approach to solve a problem in a non-repetitive stochastic environment. To accomplish this goal two research questions were asked:

1. Can an expert systems approach be applied to problems that are characterized by a non-repetitive stochastic environment?
2. Can a developed expert system prototype be interfaced with an external system in order to provide the required inputs for processing and analysis?

The first research question was answered by applying an expert systems approach to a weapons employment process model. The model development led to the establishment of a set of rules based upon a deterministic sequence of events in a controlled or bounded environment. An expert system prototype, 'EXPERTAIR', was then developed using database files which represented changing air defense scenarios. 'EXPERTAIR' was found to produce accurate expert system conclusions based upon the data input. Although the rules in 'EXPERTAIR' make the system 'deterministic', the actual weapons employment process, itself stochastic in nature, may be evaluated using a combination or different sequencing of these rules. An expansion of the weapons employment process model and further development of 'EXPERTAIR's rule base could lead to a feasible system for solving problems and recommending solutions in a stochastic environment..

Our second research question dealt with interfacing a developed expert system prototype with a system existing external and independent of the prototype. To analyze this question, DBASE III Plus was used as the external system. A database file was built using DBASE III Plus and was interfaced with 'EXPERTAIR'. The prototype was successful in analyzing and processing the information contained in the external file and was able to ascertain a recommended course of action. This demonstrated that expert systems might be interfaced and be made operable with external systems (such as a TAOC) which gives credibility to the near-term implementation of an expert system.

APPENDIX A
FRONT END FILE LISTING

```
start
        :-    nl, repeat,
          ['convert.pl'],convert,
                dbfile(Name),
                ifthenelse(Name='eof.db',stop,continu(Name)).

  continu(Name):-
                              [Name],[!message!],
          [!run_once!],
            gc(full), nl,
                  abolish(status/8),
                  abolish(target/14),
                  abolish(fighter/12),
                  retract(dbfile(Name)),
                  fail.

run_once:-
            root_instance(airdefense,I,N),
            eval(neutr_action,airdefense,I,Val,true,CF),

            fail.
run_once:- nl, nl.
stop:- nl,write('This ends the scenario.').

dbfile('xpt1.db').
dbfile('xpt2.db').
dbfile('xpt3.db').
dbfile('xpt4.db').
dbfile('xpt5.db').
dbfile('xpt6.db').
dbfile('xpt7.db').
dbfile('xpt8.db').
dbfile('xpt9.db').
dbfile('xpt10.db').
dbfile('xpt11.db').
dbfile('xpt12.db').
dbfile('xpt13.db').
dbfile('xpt14.db').
dbfile('xpt15.db').
dbfile('eof.db').

message:-
(status(DC,WC,RS,AA,TS,HS,HM,SM);
  status([DC,WC,RS,AA,TS,HS,HM,SM])).
(target(TN1,BR1,RG1,SZ1,SP1,CL1,MV1, TN2,BR2,RG2,SZ2,
                SP2,CL2,MV2);
target([[TN1,BR1,RG1,SZ1,SP1,CL1,MV1],[TN2,BR2,RG2,SZ2,
              SP2,CL2,MV2--)),
(fighter(ST1,FS1,LO1,AM1,RT1,AT1,ST2,FS2,LO2,AM2,RT2,AT2);
fighter([[ST1,FS1,LO1,AM1,RT1,AT1],[ST2,FS2,LO2,AM2,RT2,AT2]])),

          nl,write('*********** CURRENT AAW STATUS ***********'),

          nl,write('DEFENSE CONDITION : '),
          write(DC),
          write('       '),
          write('WEAPON CONDITION : '),
          write(WC),
          nl,write('RUNWAY STATUS : '),
          write(RS),
          write('       '),
```

```
write('AVAILABLE FIGHTER AIRCRAFT : '),
write(AA),
nl,write('TAOC STATUS : '),
write(TS),
write('      '),
write('HAWK STATUS : '),
write(HS),
nl,write('HAWK MISSILE INVENTORY : '),
write(HM),
write('    '),
write('STINGER MISSILE INVENTORY : '),
write(SM),

nl,write($***********  TARGET INFORMATION ***********$),

nl,write('TRACK NUMBER : '),
write(TN1),
write('      '),
write('TRACK NUMBER : '),
write(TN2),
nl,write('BEARING : '),
write(BR1),
write('           '),
write('BEARING : '),
write(BR2),
nl,write('RANGE : '),
write(RG1),
write('           '),
write('RANGE : '),
write(RG2),
nl,write('SIZE : '),
write(SZ1),
write('               '),
write('SIZE : '),
write(SZ2),
nl,write('SPEED : '),
write(SP1),
write('           '),
write('SPEED : '),
write(SP2),
nl,write('CLASS : '),
write(CL1),
write('         '),
write('CLASS : '),
write(CL2),
nl,write('MOVEMENT : '),
write(MV1),
write('      '),
write('MOVEMENT : '),
write(MV2),

nl,write('***********  FIGHTER INFORMATION  **********'),

nl,write('STATUS : '),
write(ST1),
write('               '),
write('STATUS : '),
write(ST2),
nl,write('FLT SIZE : '),
write(FS1),
write('                '),
write('FLT SIZE : '),
write(FS2),
nl,write('LOCATION : '),
write(LO1),
write('             '),
write('LOCATION : '),
write(LO2),
nl,write('AVAIL MSLS : '),
```

```
write(AM1),
write('                    '),
write('AVAIL MSLS : '),
write(AM2),
nl,write('RANGE TO TAR_TN : '),
write(RT1),
write('               '),
write('RANGE TO TAR_TN : '),
write(RT2),
nl,write('ASSIGNED TO TN : '),
write(AT1),
write('               '),
write('ASSIGNED TO TN : '),
write(AT2).
```

APPENDIX B

'EXPERTAIR' CONVERT PROGRAM LISTING

```
convert:-

[dbasefor,'prdb.pl'],
store_dbf('status1.dbf','status1.pl'),
['status1.pl'],
record(_,[DC,WC,RS,AA,TS,HS,HM,SM]),
create(Handle1,'xpt1.db'),
write(Handle1,status([DC,WC,RS,AA,TS,HS,HM,SM])),
write(Handle1,'.'), nl(Handle1),
abolish(record/2),

store_dbf('target1.dbf','target1.pl'),
['target1.pl'],
findall(X,record(Y,X),L),
nl,write(Handle1,target(L)),
write(Handle1,'.'), nl(Handle1),
abolish(record/2),

store_dbf('fighter1.dbf','fighter1.pl'),
['fighter1.pl'],
findall(X,record(Y,X),M),
nl,write(Handle1,fighter(M)),
write(Handle1,'.'), nl(Handle1),
close(Handle1),
abolish(record/2).
```

APPENDIX C

TAXONOMY FILE LISTING

```
/*--------------------------------------
            Type declarations
--------------------------------------*/

type system_status_info = role.
type target_info = role.
type fighter_info = role.
type airborne_info = role.
type intercept_info = role.

type defcon = [white,yellow,red].
type wepcon = [tight,free].
type runw_stat = [op,nonop].
type acft_avail = numeric.
type taoc_stat = [op,deg,nonop].
type hawk_stat = [op,deg,nonop,assign].
type hawk_msl = numeric.
type sting_msl = numeric.

type tar1_tn = numeric.
type tar1_bear = numeric.
type tar1_range = numeric.
type tar1_size = numeric.
type tar1_speed = numeric.
type tar1_class = [friend,asfriend,unknown,asenemy,hostile].
type tar1_move =
   [unknown,inbound,outbound,closing,orbiting,opening].
type tar2_tn = numeric.
type tar2_bear = numeric.
type tar2_range = numeric.
type tar2_size = numeric.
type tar2_speed = numeric.
type tar2_class = [friend,asfriend,unknown,asenemy,hostile].
type tar2_move =
   [unknown,inbound,outbound,closing,orbiting,opening].

type fit1_stat = [fullop,deg].
type fit1_size = numeric.
type fit1_loc = [onrun,en_cap,acap,bcap,ccap,assign].
type fit1_msl = numeric.
type fit1_range_tar = numeric.
type fit1_assign_tn = numeric.
type fit2_stat = [fullop,deg].
type fit2_size = numeric.
type fit2_loc = [onrun,en_cap,acap,bcap,ccap,assign].
type fit2_msl = numeric.
type fit2_range_tar = numeric.
type fit2_assign_tn = numeric.

type airborne_tar = [zero,one,two].
type airborne_fit = [zero,one,two].

type prob11_intercept = [zero,low,med,high].
type prob21_intercept = [zero,low,med,high].
type prob12_intercept = [zero,low,med,high].
type prob22_intercept = [zero,low,med,high].

type neutr_action = [csv,cst,cmta,cmte,cmhe,cwf11,cwf12,cwf21,cwf22,
   cf1122,cf1221,ewf11,ewf12,ewf21,ewf22,ewh1,ewh12,ews,ef1122,ef1221,
   lf1,lf12,xfa].
```

```
/*-------------------------------------------
              Concept definitions
-------------------------------------------*/

define primitive airdefense with
    system_status_info = status and
    target_info = target and
    fighter_info = fighter and
    airborne_info = airborne and
    intercept_info = intercept and
    neutr_action = [ csv,cst,cmta,cmte,cmhe,cwf11,cwf12,cwf21,cwf22,
        cf1122,cf1221,ewf11,ewf12,ewf21,ewf22,ewh1,ewh12,ews,ef1122,
        ef1221,lf1,lf12,xfa].

define primitive status with
    defcon = [white,yellow,red] and
    wepcon = [tight,free] and
    runw_stat = [op,nonop] and
    acft_avail = (0,9) and
    taoc_stat = [op,deg,nonop] and
    hawk_stat = [op,deg,nonop,assign] and
    hawk_msl = (0,15) and
    sting_msl = (0,24).

define primitive target with
    tar1_tn = (4000,4777) and
    tar1_bear = (0,359) and
    tar1_range = (0,250) and
    tar1_size = (0,12) and
    tar1_speed = (0,999) and
    tar1_class = [friend,asfriend,unknown,asenemy,hostile] and
    tar1_move =
        [unknown,inbound,outbound,closing,orbiting,opening] and
    tar2_tn = (4000,4777) and
    tar2_bear = (0,359) and
    tar2_range = (0,250) and
    tar2_size = (0,12) and
    tar2_speed = (0,999) and
    tar2_class = [friend,asfriend,unknown,asenemy,hostile] and
    tar2_move =
        [unknown,inbound,outbound,closing,orbiting,opening].

define primitive fighter with
    fit1_stat = [fullop,deg] and
    fit1_size = (0,3) and
    fit1_loc = [onrun,en_cap,acap,bcap,ccap,assign] and
    fit1_msl = (0,8) and
    fit1_range_tar = (0,250) and
    fit1_assign_tn = (4000,4777) and
    fit2_stat = [fullop,deg] and
    fit2_size = (0,3) and
    fit2_loc = [onrun,en_cap,acap,bcap,ccap,assign] and
    fit2_msl = (0,8) and
    fit2_range_tar = (0,250) and
    fit2_assign_tn = (4000,4777).

define primitive airborne with
    airborne_tar = [zero,one,two] and
    airborne_fit = [zero,one,two].

define primitive intercept with
    prob11_intercept = [zero,low,med,high] and
    prob21_intercept = [zero,low,med,high] and
    prob12_intercept = [zero,low,med,high] and
    prob22_intercept = [zero,low,med,high].
```

APPENDIX D
RULE BASE FILE LISTING

```
/*------------------------------------
            Control options
------------------------------------*/

report(neutr_action of airdefense) = $The recommendation is:$.
order(neutr_action of airdefense) = [r].
name(system_status_info of airdefense) = ignore.
name(target_info of airdefense) = ignore.
name(fighter_info of airdefense) = ignore.
name(airborne_info of airdefense) = ignore.
name(intercept_info of airdefense) = ignore.

precalc(status) =
[defcon,wepcon,runw_stat,acft_avail,taoc_stat,hawk_stat,
hawk_msl,sting_msl].

question(defcon of status) = find_defcon.
question(wepcon of status) = find_wepcon.
question(runw_stat of status) = find_runw_stat.
question(acft_avail of status) = find_acft_avail.
question(taoc_stat of status) = find_taoc_stat.
question(hawk_stat of status) = find_hawk_stat.
question(hawk_msl of status) = find_hawk_msl.
question(sting_msl of status) = find_sting_msl.

set(system_status_info of the airdefense) = system_status_info.
set(target_info of the airdefense) = target_info.
set(fighter_info of the airdefense) = fighter_info.
set(airborne_info of the airdefense) = airborne_info.
set(intercept_info of the airdefense) = intercept_info.

precalc(target) = [tar1_tn,tar1_bear,tar1_range,tar1_size,
tar1_speed,tar1_class,tar1_move,tar2_tn,tar2_bear,tar2_range,
tar2_size,tar2_speed,tar2_class,tar2_move].

question(tar1_tn of target) = find_tar1_tn.
question(tar1_bear of target) = find_tar1_bear.
question(tar1_range of target) = find_tar1_range.
question(tar1_size of target) = find_tar1_size.
question(tar1_speed of target) = find_tar1_speed.
question(tar1_class of target) = find_tar1_class.
question(tar1_move of target) = find_tar1_move.
question(tar2_tn of target) = find_tar2_tn.
question(tar2_bear of target) = find_tar2_bear.
question(tar2_range of target) = find_tar2_range.
question(tar2_size of target) = find_tar2_size.
question(tar2_speed of target) = find_tar2_speed.
question(tar2_class of target) = find_tar2_class.
question(tar2_move of target) = find_tar2_move.

precalc(fighter) = [fit1_stat,fit1_size,fit1_loc,fit1_msl,
fit1_range_tar,fit1_assign_tn,fit2_stat,fit2_size,fit2_loc,
fit2_msl,fit2_range_tar,fit2_assign_tn].

question(fit1_stat of fighter) = find_fit1_stat.
question(fit1_size of fighter) = find_fit1_size.
question(fit1_loc of fighter) = find_fit1_loc.
question(fit1_msl of fighter) = find_fit1_msl.
question(fit1_range_tar of fighter) = find_fit1_range_tar.
question(fit1_assign_tn of fighter) = find_fit1_assign_tn.
question(fit2_stat of fighter) = find_fit2_stat.
```

```
question(fit2_size of fighter) = find_fit2_size.
question(fit2_loc of fighter) = find_fit2_loc.
question(fit2_msl of fighter) = find_fit2_msl.
question(fit2_range_tar of fighter) = find_fit2_range_tar.
question(fit2_assign_tn of fighter) = find_fit2_assign_tn.

/*------------------------------------------
                  Synonyms
------------------------------------------*/

%neutr_action
synonym(csv) = $Continue normal surveillance; no immediate
               threat is present; no decision is required.$.
synonym(cst) = $Continue surveillance of target/s.  Launch (as
               required) to position fighters at CAP's A & B.  If
               a hostile act is committed, then engage target.$.
synonym(cmta) = $Continue to monitor target assignment/s.  If
               a hostile act is committed, then engage target/s.
               Update HAWK and Stinger units.$.
synonym(cmte) = $Continue to monitor target engagement/s.  If
               no kill prior to entrance to MEZ, break off flight
               assignment and engage with HAWK.   Update
               Stinger units.$.
synonym(cmhe) = $Continue to monitor HAWK target engagement.
               Update Stinger units on situation.  Be prepared to
               flush airfield.$.
synonym(cwf11) = $Cover threat (target 1) with fighter #1 and
               attempt to determine intentions.  If a hostile
               act is committed or if target enters the restricted
               area, then engage target.  Inform HAWK and Stinger
               units of situation.$.
synonym(cwf12) = $Cover threat (target 2) with fighter #1 and
               attempt to determine intentions.  If a hostile
               act is committed or if target enters the restricted
               area, then engage target.  Inform HAWK and Stinger
               units of situation.$.
synonym(cwf21) = $Cover threat (target 1) with fighter #2 and
               attempt to determine intentions.  If a hostile
               act is committed or if target enters the restricted
               area, then engage target.  Inform HAWK and Stinger
               units of situation.$.
synonym(cwf22) = $Cover threat (target 2) with fighter #2 and
               attempt to determine intentions.  If a hostile
               act is committed or if target enters the restricted
               area, then engage target.  Inform HAWK and Stinger
               units of situation.$.
synonym(cf1122) = $Cover threat (target 1) with fighter #1 and
               (target 2) with fighter #2 and attempt to determine
               intentions.  If a hostile act is committed or if
               target enters the restricted area, then engage target.
               Inform HAWK and Stinger units of situation.$.
synonym(cf1221) = $Cover threat (target 2) with fighter #1 and
               (target 1) with fighter #2 and attempt to determine
               intentions.  If a hostile act is committed or if
               target enters the restricted area, then engage target.
               Inform HAWK and Stinger units of situation.$.
synonym(ewf11) = $Attempt to engage threat (target 1) with
               fighter #1.  If no kill prior to entrance
               to MEZ, break off assignment and engage with
               HAWK.  Inform Stinger units of situation.$.
synonym(ewf12) = $Attempt to engage threat (target 2) with
               fighter #1.  If no kill prior to entrance
               to MEZ, break off assignment and engage with
               HAWK.  Inform Stinger units of situation.$.
synonym(ewf21) = $Attempt to engage threat (target 1) with
               fighter #2.  If no kill prior to entrance
               to MEZ, break off assignment and engage with
               HAWK.  Inform Stinger units of situation.$.
synonym(ewf22) = $Attempt to engage threat (target 2) with
```

```
                  fighter #2.  If no kill prior to entrance
                  to MEZ, break off assignment and engage with
                  HAWK.  Inform Stinger units of situation.$.
synonym(ewh1) = $Attempt to engage threat (target 1) with HAWK
                  missile when target is in range.  Inform
                  Stinger units of situation.$.
synonym(ef1122) = $Attempt to engage threat (target 1) with
                  fighter #1 and (target 2) with fighter #2.  If
                  no kill prior to entrance to MEZ, break off
                  assignment and engage with HAWK.  Inform
                  Stinger units of situation.$.
synonym(ef1221) = $Attempt to engage threat (target 2) with
                  fighter #1 and (target 1) with fighter #2.  If
                  no kill prior to entrance to MEZ, break off
                  assignment and engage with HAWK.  Inform
                  Stinger units of situation.$.
synonym(ewh12) = $Attempt to engage threat (targets 1 & 2) with
                  HAWK missile when targets are in range.
                  Inform Stinger units of situation.  Be
                  prepared to flush airfield.$.
synonym(ews) = $Attempt to engage target/s with Stinger missile
                  units.  Be prepared to flush airfield.$.
synonym(lf1) = $Launch fighter #1 and assign to Cap C.
                  Attempt to determine intentions of target.
                  If target is identified as a possible threat,
                  set Defense Condition to Yellow.$.
synonym(lf12) = $Launch fighter #1 and assign to Cap A.
                  Launch fighter #2 and assign to Cap B.  Attempt
                  to determine intentions of target.  If target
                  is identified as a possible threat, set Defense
                  Condition to Yellow.$.
synonym(xfa) = $Cancel cover/engage assignment.  Target is no
                  longer an immediate threat.  Return fighters to
                  CAP or return to base and replace as required.$.

%Rule 1
the airborne_tar of airborne_info is zero
         if
the tar1_size of target_info is X and
         X < 1 and
the tar2_size of target_info is Y and
         Y < 1.

%Rule 2
the airborne_tar of airborne_info is zero
         if
the tar1_tn of target_info is X and
         X < 1 and
the tar2_tn of target_info is Y and
         Y < 1.

%Rule 3
the airborne_tar of airborne_info is one
         if
the tar1_size of target_info is X and
         X > 1 and
the tar2_size of target_info is Y and
         Y < 1.

%Rule 4
the airborne_tar of airborne_info is one
         if
the tar1_tn of target_info is X and
         X > 1 and
the tar2_tn of target_info is Y and
         Y < 1.

%Rule 5
the airborne_tar of airborne_info is two
```

```
        if
the tar1_tn of target_info is X and
        X > 1 and
the tar2_tn of target_info is Y and
        Y > 1.

%Rule 6
the airborne_tar of airborne_info is two
        if
the tar1_size of target_info is X and
        X > 1 and
the tar2_size of target_info is Y and
        Y > 1.

%Rule 7
the airborne_fit of airborne_info is zero
        if
the fit1_loc of fighter_info is onrun and
the fit2_loc of fighter_info is onrun.

%Rule 8
the airborne_fit of airborne_info is one
        if
the fit1_loc of fighter_info is
        [en_cap,acap,ccap,assign] and
the fit2_loc of fighter_info is onrun.

%Rule 9
the airborne_fit of airborne_info is two
        if
the fit1_loc of fighter_info is
        [en_cap,acap,assign] and
the fit2_loc of fighter_info is
        [en_cap,bcap,assign].

%Rule 10
the prob11_intercept of intercept_info is zero
        if
the tar1_size of target_info is X and
        X < 1.

%Rule 11
  the prob11_intercept of intercept_info is zero
        if
  the runw_stat of system_status_info is nonop and
  the fit1_loc of fighter_info is onrun.

%Rule 12
  the prob11_intercept of intercept_info is zero
        if
  the tar1_range of target_info is X and
        X < 10 and
  the tar1_speed of target_info is Y and
        Y > 350 and
  the tar1_move of target_info is inbound and
  the fit1_loc of fighter_info is onrun.

%Rule 13
the prob21_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 10 and
the tar1_speed of target_info is Y and
        Y > 350 and
the tar1_move of target_info is inbound and
the fit2_loc of fighter_info is onrun.

%Rule 14
  the prob11_intercept of intercept_info is zero
```

```
        if
  the acft_avail of system_status_info is X and
        X < 1.

%Rule 15
the prob21_intercept of intercept_info is zero
        if
the acft_avail of system_status_info is X and
        X < 1.

%Rule 16
  the prob11_intercept of intercept_info is zero
        if
  the fit1_msl of fighter_info is X and
        X < 1.

%Rule 17
the prob21_intercept of intercept_info is zero
        if
the fit2_msl of fighter_info is X and
        X < 1.

%Rule 18
the prob11_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 65 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_loc of fighter_info is [acap,ccap].

%Rule 19
the prob21_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 65 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_loc of fighter_info is acap and
the fit2_loc of fighter_info is bcap.

%Rule 20
the prob11_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 90 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is acap and
the fit1_range_tar of fighter_info is Z and
        Z > 20 and
        Z < 50.

%Rule 21
the prob21_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 90 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is [inbound,closing] and
the fit2_loc of fighter_info is bcap and
the fit2_range_tar of fighter_info is Z and
        Z > 20 and
        Z < 50.
```

```
%Rule 22
the prob11_intercept of intercept_info is zero
        if
the tar1_range of target_info is X and
        X < 70 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is ccap and
the fit1_range_tar of fighter_info is Z and
        Z > 20 and
        Z < 50.

%Rule 23
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 10 and
        X < 45 and
the tar1_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is onrun.

%Rule 24
the prob21_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 10 and
        X < 25 and
the tar1_move of target_info is [inbound,closing] and
the fit2_loc of fighter_info is onrun.

%Rule 25
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 64 and
        X < 80 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_loc of fighter_info is ccap and
the fit1_range_tar of fighter_info is Z and
        Z < 21.

%Rule 26
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 69 and
        X < 80 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_loc of fighter_info is ccap and
the fit1_range_tar of fighter_info is Z and
        Z > 20 and
        Z < 50.

%Rule 27
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 64 and
        X < 100 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_loc of fighter_info is acap and
the fit1_range_tar of fighter_info is Z and
```

```
        Z < 21.

%Rule 28
the prob21_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 64 and
        X < 100 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit2_loc of fighter_info is bcap and
the fit2_range_tar of fighter_info is Z and
        Z < 21.

%Rule 29
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 79 and
        X < 100 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_stat of fighter_info is deg and
the fit1_loc of fighter_info is ccap.

%Rule 30
the prob11_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 99 and
        X < 120 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit1_stat of fighter_info is deg and
the fit1_loc of fighter_info is acap.

%Rule 31
the prob21_intercept of intercept_info is low
        if
the tar1_range of target_info is X and
        X > 99 and
        X < 120 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit2_stat of fighter_info is deg and
the fit2_loc of fighter_info is bcap.

%Rule 32
the prob21_intercept of intercept_info is med
        if
the tar1_bear of target_info is W and
        W > 70 and
        W < 140 and
the tar1_range of target_info is X and
        X > 109 and
        X < 120 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is inbound and
the fit2_stat of fighter_info is fullop and
the fit2_loc of fighter_info is bcap.

%Rule 33
the prob22_intercept of intercept_info is med
        if
```

```
the tar2_bear of target_info is W and
        W > 70 and
        W < 140 and
the tar2_range of target_info is X and
        X > 109 and
        X < 120 and
the tar2_speed of target_info is Y and
        Y > 400 and
the tar2_move of target_info is [inbound,closing] and
the fit2_stat of fighter_info is fullop and
the fit2_loc of fighter_info is bcap.

%Rule 34
the prob11_intercept of intercept_info is med
        if
the tar1_bear of target_info is W and
        W < 70 and
the tar1_range of target_info is X and
        X > 109 and
        X < 120 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is acap.

%Rule 35
the prob12_intercept of intercept_info is med
        if
the tar2_bear of target_info is W and
        W < 70 and
the tar2_range of target_info is X and
        X > 109 and
        X < 120 and
the tar2_speed of target_info is Y and
        Y > 400 and
the tar2_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is acap.

%Rule 36
the prob11_intercept of intercept_info is med
        if
the tar1_range of target_info is X and
        X > 90 and
        X < 110 and
the tar1_speed of target_info is Y and
        Y > 400 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is ccap.

%Rule 37
the prob12_intercept of intercept_info is med
        if
the tar2_range of target_info is X and
        X > 90 and
        X < 110 and
the tar2_speed of target_info is Y and
        Y > 400 and
the tar2_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is ccap.

%Rule 38
the prob11_intercept of intercept_info is med
        if
the tar1_bear of target_info is X and
        X < 70 and
```

```
the tar1_range of target_info is Y and
        Y > 119 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is deg and
the fit1_loc of fighter_info is acap.

%Rule 39
the prob12_intercept of intercept_info is med
        if
the tar2_bear of target_info is X and
        X < 70 and
the tar2_range of target_info is Y and
        Y > 119 and
the tar2_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is deg and
the fit1_loc of fighter_info is acap.

%Rule 40
the prob21_intercept of intercept_info is med
        if
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_range of target_info is Y and
        Y > 119 and
the tar1_move of target_info is [inbound,closing] and
the fit2_stat of fighter_info is deg and
the fit2_loc of fighter_info is bcap.

%Rule 41
the prob22_intercept of intercept_info is med
        if
the tar2_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar2_range of target_info is Y and
        Y > 119 and
the tar2_move of target_info is [inbound,closing] and
the fit2_stat of fighter_info is deg and
the fit2_loc of fighter_info is bcap.

%Rule 42
the prob11_intercept of intercept_info is med
        if
the tar1_range of target_info is X and
        X > 109 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is deg and
the fit1_loc of fighter_info is ccap.

%Rule 43
the prob11_intercept of intercept_info is high
        if
the tar1_bear of target_info is X and
        X < 70 and
the tar1_range of target_info is Y and
        Y > 119 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is
        [onrun,en_cap,acap].

%Rule 44
the prob12_intercept of intercept_info is high
        if
the tar2_bear of target_info is X and
        X < 70 and
the tar2_range of target_info is Y and
        Y > 119 and
```

```
the tar2_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is
        [onrun,en_cap,acap].

%Rule 45
the prob21_intercept of intercept_info is high
        if
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_range of target_info is Y and
        Y > 119 and
the tar1_move of target_info is [inbound,closing] and
the fit2_stat of fighter_info is fullop and
the fit2_loc of fighter_info is
        [onrun,en_cap,bcap].

%Rule 46
the prob22_intercept of intercept_info is high
        if
the tar2_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar2_range of target_info is Y and
        Y > 119 and
the tar2_move of target_info is [inbound,closing] and
the fit2_stat of fighter_info is fullop and
the fit2_loc of fighter_info is
        [onrun,en_cap,bcap].

%Rule 47
the prob11_intercept of intercept_info is high
        if
the tar1_range of target_info is X and
        X > 109 and
the tar1_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is
        [onrun,en_cap,ccap].

%Rule 48
the prob12_intercept of intercept_info is high
        if
the tar2_range of target_info is X and
        X > 109 and
the tar2_move of target_info is [inbound,closing] and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is
        [onrun,en_cap,ccap].

%Rule 49
the prob11_intercept of intercept_info is high
        if
the tar1_range of target_info is X and
        X < 200 and
the tar1_move of target_info is orbiting and
the fit1_stat of fighter_info is fullop and
the fit1_loc of fighter_info is [en_cap,acap,ccap].

%Rule 50
the neutr_action of airdefense is csv
        if
the defcon of system_status_info is white and
the airborne_tar of airborne_info is zero.

%Rule 51
the neutr_action of airdefense is csv
        if
```

```
the defcon of system_status_info is white and
the tar1_class of target_info is [friend,asfriend] and
the airborne_tar of airborne_info is one.

%Rule 52
the neutr_action of airdefense is cmta
        if
the tar1_class of target_info is [unknown,asenemy] and
the tar1_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is assign and
the airborne_tar of airborne_info is one.

%Rule 53
the neutr_action of airdefense is cmta
        if
the tar1_class of target_info is [unknown,asenemy] and
the tar1_move of target_info is [inbound,closing] and
the tar2_class of target_info is [unknown,asenemy] and
the tar2_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is assign and
the fit2_loc of fighter_info is assign.

%Rule 54
the neutr_action of airdefense is cmte
        if
the tar1_class of target_info is hostile and
the tar1_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is assign and
the airborne_tar of airborne_info is one.

%Rule 55
the neutr_action of airdefense is cmte
        if
the tar1_class of target_info is hostile and
the tar1_move of target_info is [inbound,closing] and
the tar2_class of target_info is hostile and
the tar2_move of target_info is [inbound,closing] and
the fit1_loc of fighter_info is assign and
the fit2_loc of fighter_info is assign.

%Rule 56
the neutr_action of airdefense is cmhe
        if
the hawk_stat of system_status_info is assign and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar1_move of target_info is [inbound,closing] and
the airborne_tar of airborne_info is one.

%Rule 57
the neutr_action of airdefense is lf1
        if
the defcon of system_status_info is white and
the runw_stat of system_status_info is op and
the acft_avail of system_status_info is X and
        X > 0 and
the airborne_tar of airborne_info is one.

%Rule 58
the neutr_action of airdefense is lf12
        if
the defcon of system_status_info is white and
the runw_stat of system_status_info is op and
the acft_avail of system_status_info is X and
        X > 1 and
the airborne_tar of airborne_info is two.

%Rule 59
the neutr_action of airdefense is ewh1
        if
```

```
the wepcon of system_status_info is free and
the runw_stat of system_status_info is nonop and
the hawk_stat of system_status_info is [op,deg] and
the hawk_msl of system_status_info is W and
        W > 0 and
the tar1_bear of target_info is X and
        X > 0 and
        X < 140 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the airborne_fit of airborne_info is zero.

%Rule 60
the neutr_action of airdefense is ewh1
        if
the hawk_stat of system_status_info is [op,deg] and
the hawk_msl of system_status_info is X and
        X > 0 and
the tar1_class of target_info is hostile and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is zero.

%Rule 61
the neutr_action of airdefense is ewh1
        if
the hawk_stat of system_status_info is [op,deg] and
the hawk_msl of system_status_info is X and
        X > 0 and
the tar1_class of target_info is hostile and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is two and
the prob11_intercept of intercept_info is zero and
the prob21_intercept of intercept_info is zero.

%Rule 62
the neutr_action of airdefense is ewh12
        if
the hawk_stat of system_status_info is [op,deg] and
the hawk_msl of system_status_info is X and
        X > 1 and
the tar1_class of target_info is hostile and
the tar2_class of target_info is hostile and
the airborne_tar of airborne_info is two and
the airborne_fit of airborne_info is two and
the prob11_intercept of intercept_info is zero and
the prob21_intercept of intercept_info is zero and
the prob12_intercept of intercept_info is zero and
the prob22_intercept of intercept_info is zero.

%Rule 63
the neutr_action of airdefense is ews
        if
the hawk_stat of system_status_info is nonop and
the sting_msl of system_status_info is X and
        X > 0 and
the tar1_class of target_info is hostile and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is zero.

%Rule 64
the neutr_action of airdefense is ews
        if
the hawk_msl of system_status_info is X and
        X < 1 and
the sting_msl of system_status_info is Y and
        Y > 0 and
the tar1_class of target_info is hostile and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is zero.
```

```
%Rule 65
the neutr_action of airdefense is cst
        if
the defcon of system_status_info is yellow and
the runw_stat of system_status_info is op and
the acft_avail of system_status_info is X and
        X > 1 and
the tar1_class of target_info is [asenemy,unknown] and
the tar1_move of target_info is [outbound,orbiting,opening] and
the tar2_class of target_info is [asenemy,unknown] and
the tar2_move of target_info is [outbound,orbiting,opening].

%Rule 66
the neutr_action of airdefense is xfa
        if
the tar1_class of target_info is [asenemy,unknown] and
the tar1_move of target_info is [outbound,opening] and
the fit1_range_tar of fighter_info is X and
        X > 25 and
the airborne_tar of airborne_info is one.

%Rule 67
the neutr_action of airdefense is xfa
        if
the tar1_class of target_info is [asenemy,unknown] and
the tar1_move of target_info is [outbound,opening] and
the fit2_range_tar of fighter_info is X and
        X > 25 and
the airborne_tar of airborne_info is one.

%Rule 68
the neutr_action of airdefense is xfa
        if
the tar1_class of target_info is [asenemy,unknown] and
the tar1_move of target_info is [outbound,opening] and
the tar2_class of target_info is [asenemy,unknown] and
the tar2_move of target_info is [outbound,opening] and
the fit1_range_tar of fighter_info is X and
        X > 25 and
the fit2_range_tar of fighter_info is Y and
        Y > 25.

%Rule 69
the neutr_action of airdefense is cwf11
        if
the wepcon of system_status_info is tight and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [asenemy,unknown] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is [med,high].

%Rule 70
the neutr_action of airdefense is cwf11
        if
the wepcon of system_status_info is tight and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [asenemy,unknown] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is two and
the prob11_intercept of intercept_info is [med,high].

%Rule 71
the neutr_action of airdefense is cwf21
        if
the wepcon of system_status_info is tight and
```

```
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [asenemy,unknown] and
the tar2_class of target_info is [asenemy,unknown] and
the tar2_move of target_info is [outbound,orbiting,opening] and
the prob21_intercept of intercept_info is [med,high].

%Rule 72
the neutr_action of airdefense is cwf21
        if
the wepcon of system_status_info is tight and
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [asenemy,unknown] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is two and
the prob21_intercept of intercept_info is [med,high].

%Rule 73
the neutr_action of airdefense is cwf12
        if
the wepcon of system_status_info is tight and
the tar1_move of target_info is [outbound,orbiting,opening] and
the tar2_class of target_info is [asenemy,unknown] and
the airborne_fit of airborne_info is one and
the prob12_intercept of intercept_info is [med,high].

%Rule 74
the neutr_action of airdefense is cwf12
        if
the wepcon of system_status_info is tight and
the tar1_move of target_info is [outbound,orbiting,opening] and
the tar2_bear of target_info is X and
        X < 70 and
the tar2_class of target_info is [asenemy,unknown] and
the airborne_fit of airborne_info is two and
the prob12_intercept of intercept_info is [med,high].

%Rule 75
the neutr_action of airdefense is cf1221
        if
the wepcon of system_status_info is tight and
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [asenemy,unknown] and
the tar2_bear of target_info is Y and
        Y < 70 and
the tar2_class of target_info is [asenemy,unknown] and
the prob12_intercept of intercept_info is [med,high] and
the prob21_intercept of intercept_info is [med,high].

%Rule 76
the neutr_action of airdefense is cf1122
        if
the wepcon of system_status_info is tight and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [asenemy,unknown] and
the tar2_bear of target_info is Y and
        Y > 70 and
        Y < 140 and
the tar2_class of target_info is [asenemy,unknown] and
the prob11_intercept of intercept_info is [med.high] and
the prob22_intercept of intercept_info is [med,high].

%Rule 77
```

```
the neutr_action of airdefense is ewf11
        if
the wepcon of system_status_info is free and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is [med,high].

%Rule 78
the neutr_action of airdefense is ewf11
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is two and
the prob11_intercept of intercept_info is [med,high].

%Rule 79
the neutr_action of airdefense is ewf11
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar2_move of target_info is [outbound,opening] and
the airborne_fit of airborne_info is two and
the prob11_intercept of intercept_info is [med,high].

%Rule 80
the neutr_action of airdefense is ewf11
        if
the wepcon of system_status_info is free and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar2_move of target_info is [outbound,opening] and
the airborne_fit of airborne_info is one and
the prob11_intercept of intercept_info is [med,high].

%Rule 81
the neutr_action of airdefense is ewf21
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the airborne_tar of airborne_info is one and
the airborne_fit of airborne_info is two and
the prob21_intercept of intercept_info is [med,high].

%Rule 82
the neutr_action of airdefense is ewf21
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar2_move of target_info is [outbound,opening] and
the airborne_fit of airborne_info is two and
the prob21_intercept of intercept_info is [med,high].

%Rule 83
the neutr_action of airdefense is ewf12
        if
the wepcon of system_status_info is free and
the tar1_move of target_info is [outbound,opening] and
the tar2_bear of target_info is X and
```

```
        X < 70 and
the tar2_class of target_info is [unknown,asenemy,hostile] and
the airborne_fit of airborne_info is two and
the prob12_intercept of intercept_info is [med,high].

%Rule 84
the neutr_action of airdefense is ewf12
        if
the wepcon of system_status_info is free and
the tar1_move of target_info is [outbound,opening] and
the tar2_class of target_info is [unknown,asenemy,hostile] and
the airborne_fit of airborne_info is one and
the prob12_intercept of intercept_info is [med,high].

%Rule 85
the neutr_action of airdefense is ewf22
        if
the wepcon of system_status_info is free and
the tar1_move of target_info is X and
        X > 70 and
        X < 140 and
the tar2_class of target_info is [unknown,asenemy,hostile] and
the airborne_fit of airborne_info is two and
the prob22_intercept of intercept_info is [med,high].

%Rule 86
the neutr_action of airdefense is ef1122
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X < 70 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar2_bear of target_info is Y and
        Y > 70 and
        Y < 140 and
the tar2_class of target_info is [unknown,asenemy,hostile] and
the prob11_intercept of intercept_info is [med,high] and
the prob22_intercept of intercept_info is [med,high].

%Rule 87
the neutr_action of airdefense is ef1221
        if
the wepcon of system_status_info is free and
the tar1_bear of target_info is X and
        X > 70 and
        X < 140 and
the tar1_class of target_info is [unknown,asenemy,hostile] and
the tar2_bear of target_info is Y and
        Y < 70 and
the tar2_class of target_info is [unknown,asenemy,hostile] and
the prob12_intercept of intercept_info is [med,high] and
the prob21_intercept of intercept_info is [med,high].
```

APPENDIX E

CALCULATIONS FILE LISTING

```
/*-------------------------------------------
              CALCULATIONS
-------------------------------------------*/

find_defcon(I,status,defcon,[DEFCON/1.0]):-
        status(DEFCON,_,_,_,_,_,_,_);
        status([DEFCON,_,_,_,_,_,_,_]).
find_wepcon(I,status,wepcon,[WEPCON/1.0]):-
        status(_,WEPCON,_,_,_,_,_,_);
        status([_,WEPCON,_,_,_,_,_,_]).
find_runw_stat(I,status,runw_stat,[RUNW_STAT/1.0]):-
        status(_,_,RUNW_STATUS,_,_,_,_,_);
        status([_,_,RUNW_STATUS,_,_,_,_,_]).
find_acft_avail(I,status,acft_avail,[ACFT_AVAIL/1.0]):-
        status(_,_,_,ACFT_AVAIL,_,_,_,_);
        status([_,_,_,ACFT_AVAIL,_,_,_,_]).
find_taoc_stat(I,status,taoc_stat,[TAOC_STAT/1.0]):-
        status(_,_,_,_,TAOC_STAT,_,_,_);
        status([_,_,_,_,TAOC_STAT,_,_,_]).
find_hawk_stat(I,status,hawk_stat,[HAWK_STAT/1.0]):-
        status(_,_,_,_,_,HAWK_STAT,_,_);
        status([_,_,_,_,_,HAWK_STAT,_,_]).
find_hawk_msl(I,status,hawk_msl,[HAWK_MSL/1.0]):-
        status(_,_,_,_,_,_,HAWK_MSL,_);
        status([_,_,_,_,_,_,HAWK_MSL,_]).
find_sting_msl(I,status,sting_msl,[STING_MSL/1.0]):-
        status(_,_,_,_,_,_,_,STING_MSL);
        status([_,_,_,_,_,_,_,STING_MSL]).

find_tar1_tn(I,target,tar1_tn,[TAR1_TN/1.0]):-
        target(TAR1_TN,_,_,_,_,_,_,_,_,_,_,_,_,_);
        target([[TAR1_TN,_,_,_,_,_,_],[_,_,_,_,_,_,_]]).
find_tar1_bear(I,target,tar1_bear,[TAR1_BEAR/1.0]):-
        target(_,TAR1_BEAR,_,_,_,_,_,_,_,_,_,_,_,_);
        target([[_,TAR1_BEAR,_,_,_,_,_],[_,_,_,_,_,_,_]]).
find_tar1_range(I,target,tar1_range,[TAR1_RANGE/1.0]):-
        target(_,_,TAR1_RANGE,_,_,_,_,_,_,_,_,_,_,_);
        target([[_,_,TAR1_RANGE,_,_,_,_],[_,_,_,_,_,_,_]]).
find_tar1_size(I,target,tar1_size,[TAR1_SIZE/1.0]):-
        target(_,_,_,TAR1_SIZE,_,_,_,_,_,_,_,_,_,_);
        target([[_,_,_,TAR1_SIZE,_,_,_],[_,_,_,_,_,_,_]]).
find_tar1_speed(I,target,tar1_speed,[TAR1_SPEED/1.0]):-
        target(_,_,_,_,TAR1_SPEED,_,_,_,_,_,_,_,_,_);
        target([[_,_,_,_,TAR1_SPEED,_,_],[_,_,_,_,_,_,_]]).
find_tar1_class(I,target,tar1_class,[TAR1_CLASS/1.0]):-
        target(_,_,_,_,_,TAR1_CLASS,_,_,_,_,_,_,_,_);
        target([[_,_,_,_,_,TAR1_CLASS,_],[_,_,_,_,_,_,_]]).
find_tar1_move(I,target,tar1_move,[TAR1_MOVE/1.0]):-
        target(_,_,_,_,_,_,TAR1_MOVE,_,_,_,_,_,_,_);
        target([[_,_,_,_,_,_,TAR1_MOVE],[_,_,_,_,_,_,_]]).
find_tar2_tn(I,target,tar2_tn,[TAR2_TN/1.0]):-
        target(_,_,_,_,_,_,_,TAR2_TN,_,_,_,_,_,_);
        target([[_,_,_,_,_,_,_],[TAR2_TN,_,_,_,_,_,_]]).
find_tar2_bear(I,target,tar2_bear,[TAR2_BEAR/1.0]):-
        target(_,_,_,_,_,_,_,_,TAR2_BEAR,_,_,_,_,_);
        target([[_,_,_,_,_,_,_],[_,TAR2_BEAR,_,_,_,_,_]]).
find_tar2_range(I,target,tar2_range,[TAR2_RANGE/1.0]):-
        target(_,_,_,_,_,_,_,_,_,TAR2_RANGE,_,_,_,_);
        target([[_,_,_,_,_,_,_],[_,_,TAR2_RANGE,_,_,_,_]]).
```

```
find_tar2_size(I,target,tar2_size,[TAR2_SIZE/1.0]):-
        target(_,_,_,_,_,_,_,_,_,TAR2_SIZE,_,_,_);
        target([[_,_,_,_,_,_,_],[_,_,TAR2_SIZE,_,_,_]]).
find_tar2_speed(I,target,tar2_speed,[TAR2_SPEED/1.0]):-
        target(_,_,_,_,_,_,_,_,_,_,TAR2_SPEED,_,_);
        target([[_,_,_,_,_,_,_],[_,_,_,TAR2_SPEED,_,_]]).
find_tar2_class(I,target,tar2_class,[TAR2_CLASS/1.0]):-
        target(_,_,_,_,_,_,_,_,_,_,_,TAR2_CLASS,_);
        target([[_,_,_,_,_,_,_],[_,_,_,_,TAR2_CLASS,_]]).
find_tar2_move(I,target,tar2_move,[TAR2_MOVE/1.0]):-
        target(_,_,_,_,_,_,_,_,_,_,_,_,TAR2_MOVE);
        target([[_,_,_,_,_,_,_],[_,_,_,_,_,TAR2_MOVE]]).

find_fit1_stat(I,fighter,fit1_stat,[FIT1_STAT/1.0]):-
        fighter(FIT1_STAT,_,_,_,_,_,_,_,_,_,_,_);
        fighter([[FIT1_STAT,_,_,_,_,_],[_,_,_,_,_,_]]).
find_fit1_size(I,fighter,fit1_size,[FIT1_SIZE/1.0]):-
        fighter(_,FIT1_SIZE,_,_,_,_,_,_,_,_,_,_);
        fighter([[_,FIT1_SIZE,_,_,_,_],[_,_,_,_,_,_]]).
find_fit1_loc(I,fighter,fit1_loc,[FIT1_LOC/1.0]):-
        fighter(_,_,FIT1_LOC,_,_,_,_,_,_,_,_,_);
        fighter([[_,_,FIT1_LOC,_,_,_],[_,_,_,_,_,_]]).
find_fit1_msl(I,fighter,fit1_msl,[FIT1_MSL/1.0]):-
        fighter(_,_,_,FIT1_MSL,_,_,_,_,_,_,_,_);
        fighter([[_,_,_,FIT1_MSL,_,_],[_,_,_,_,_,_]]).
find_fit1_range_tar(I,fighter,fit1_range_tar,
        [FIT1_RANGE_TAR/1.0]):-
        fighter(_,_,_,_,FIT1_RANGE_TAR,_,_,_,_,_,_,_);
        fighter([[_,_,_,_,FIT1_RANGE_TAR,_],[_,_,_,_,_,_]]).
find_fit1_assign_tn(I,fighter,fit1_assign_tn,
        [FIT1_ASSIGN_TN/1.0]):-
        fighter(_,_,_,_,_,FIT1_ASSIGN_TN,_,_,_,_,_,_);
        fighter([[_,_,_,_,_,FIT1_ASSIGN_TN],[_,_,_,_,_,_]]).
find_fit2_stat(I,fighter,fit2_stat,[FIT2_STAT/1.0]):-
        fighter(_,_,_,_,_,_,FIT2_STAT,_,_,_,_,_);
        fighter([[_,_,_,_,_,_],[FIT2_STAT,_,_,_,_,_]]).
find_fit2_size(I,fighter,fit2_size,[FIT2_SIZE/1.0]):-
        fighter(_,_,_,_,_,_,_,FIT2_SIZE,_,_,_,_);
        fighter([[_,_,_,_,_,_],[_,FIT2_SIZE,_,_,_,_]]).
find_fit2_loc(I,fighter,fit2_loc,[FIT2_LOC/1.0]):-
        fighter(_,_,_,_,_,_,_,_,FIT2_LOC,_,_,_);
        fighter([[_,_,_,_,_,_],[_,_,FIT2_LOC,_,_,_]]).
find_fit2_msl(I,fighter,fit2_msl,[FIT2_MSL/1.0]):-
        fighter(_,_,_,_,_,_,_,_,_,FIT2_MSL,_,_);
        fighter([[_,_,_,_,_,_],[_,_,_,FIT2_MSL,_,_]]).
find_fit2_range_tar(I,fighter,fit2_range_tar,
        [FIT2_RANGE_TAR/1.0]):-
        fighter(_,_,_,_,_,_,_,_,_,_,FIT2_RANGE_TAR,_);
        fighter([[_,_,_,_,_,_],[_,_,_,_,FIT2_RANGE_TAR,_]]).
find_fit2_assign_tn(I,fighter,fit2_assign_tn,
        [FIT2_ASSIGN_TN/1.0]):-
        fighter(_,_,_,_,_,_,_,_,_,_,_,FIT2_ASSIGN_TN);
        fighter([[_,_,_,_,_,_],[_,_,_,_,_,FIT2_ASSIGN_TN]]).
```

APPENDIX F
DATABASE FILES LISTING

```
/*------------------------------------------
            Database XPT1
------------------------------------------*/

status(white,tight,op,9,op,op,12,20).
target(0,0,0,0,0,unknown,unknown,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,1,onrun,6,0,0,fullop,1,onrun,6,0,0).

/*------------------------------------------
            Database XPT1 (converted)
------------------------------------------*/

status([white     ,tight     ,op     ,9.0,op     ,op     ,12.0,20.0]).
target([[0.0,0.0,0.0,0.0,0.0,unknown ,unknown    ],[0.0,0.0,0.0,0.0,0.0,unknown ,unknown    ]]).
fighter([[fullop ,1.0,onrun   ,6.0,0.0,0.0],[fullop ,1.0,onrun   ,6.0,0.0,0.0]]).

/*------------------------------------------
            Database XPT2
------------------------------------------*/

status(white,tight,op,9,op,op,12,20).
target(4001,135,200,1,350,asfriend,closing,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,1,onrun,6,0,0,fullop,1,onrun,6,0,0).

/*------------------------------------------
            Database XPT3
------------------------------------------*/

status(white,tight,op,9,op,op,12,20).
target(4002,110,185,1,375,asenemy,orbiting,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,1,onrun,6,0,0,fullop,1,onrun,6,0,0).

/*------------------------------------------
            Database XPT4
------------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4002,110,185,1,375,asenemy,orbiting,
            4003,028,170,2,400,asenemy,orbiting).
fighter(fullop,1,ccap,6,0,0,fullop,1,onrun,6,0,0).

/*------------------------------------------
            Database XPT5
------------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4002,110,170,1,425,asenemy,inbound,
            4003,030,190,2,400,asenemy,outbound).
fighter(fullop,1,acap,6,0,0,fullop,1,bcap,6,0,0).
```

```
/*-----------------------------------------
            Database XPT6
-----------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4002,110,175,1,425,asenemy,outbound,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,1,acap,6,0,0,
              fullop,1,assign,6,65,4002).


/*-------------------------------------------
            Database XPT7
-------------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4010,050,190,1,400,asenemy,inbound,
            4012,130,175,1,400,asenemy,inbound).
fighter(fullop,1,acap,6,0,0,fullop,1,bcap,6,0,0).

/*-----------------------------------------
            Database XPT8
-----------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4010,050,180,1,400,asenemy,inbound,
            4012,128,165,1,410,asenemy,inbound).
fighter(fullop,1,assign,6,78,4010,
              fullop,1,assign,6,57,4012).


/*-------------------------------------------
            Database XPT9
-------------------------------------------*/

status(yellow,tight,op,9,op,op,12,20).
target(4020,130,60,1,600,hostile,inbound,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,2,acap,6,0,0,fullop,2,bcap,6,0,0).

/*-----------------------------------------
            Database XPT10
-----------------------------------------*/

status(red,free,op,9,op,assign,12,20).
target(4020,130,40,1,600,hostile,inbound,
            0,0,0,0,0,unknown,unknown).
fighter(fullop,2,acap,6,0,0,fullop,2,bcap,6,0,0).

/*-------------------------------------------
            Database XPT11
-------------------------------------------*/

status(red,free,op,9,op,op,10,18).
target(4024,025,135,2,450,hostile,inbound,
            0,0,0,0,0,unknown,unknown).
fighter(deg,1,acap,4,0,0,fullop,1,bcap,6,0,0).

/*------------------------------------------
            Database XPT12
------------------------------------------*/

status(red,free,op,9,op,op,10,18).
target(4027,75,160,4,480,asenemy,inbound,
            4030,60,170,2,500,hostile,inbound).
fighter(fullop,2,acap,4,0,0,deg,1,bcap,6,0,0).
/*-------------------------------------------
            Database XPT13
-------------------------------------------*/
```

```
status(red,free,op,9,op,op,10,18).
target(4027,75,130,4,480,hostile,inbound,
             4030,62,140,2,500,hostile,inbound).
fighter(fullop,2,assign,4,25,4030,
               deg,2,assign,6,15,4027).

/*-------------------------------------------
             Database XPT14
-------------------------------------------*/

status(red,free,op,6,op,op,10,18).
target(4027,75,100,2,500,hostile,inbound,
             0,0,0,0,0,unknown,unknown).
fighter(fullop,1,acap,0,0,0,
               fullop,1,onrun,6,0,0).

/*-------------------------------------------
             Database XPT14
-------------------------------------------*/

status(red,free,op,6,op,op,10,18).
target(4027,75,100,2,500,hostile,inbound,
             0,0,0,0,0,unknown,unknown).
fighter(fullop,1,acap,0,0,0,
               fullop,1,onrun,6,0,0).

/*-------------------------------------------
             Database XPT15
-------------------------------------------*/

status(red,free,nonop,2,deg,nonop,3,6).
target(4040,38,45,4,600,hostile,inbound,
             4042,75,38,4,500,hostile,inbound).
fighter(deg,1,ccap,0,0,0,fullop,2,onrun,6,0,0).
```

APPENDIX G

'EXPERTAIR' DICTIONARY

airdefense
the current air defense posture or environment as defined by data input identifying threat level, status of support and weapon systems, and airborne target and fighter characteristics

defcon
defense condition; level of readiness based upon projected imminence of a hostile attack; white delineates no expected attack; yellow anticipates attack in the near future; red indicates attack is in progress

wepcon
weapon condition; identifies weapon system rule of engagement; tight is defined in this model as engagement only in self defense; weapons free is the engagement of all targets not identified as friend

runw_stat
runway status

acft_avail
aircraft availability; identifies the total nunber of operational aircraft

taoc_stat
tactical air operations center status

hawk_stat
IHAWK status; identifies the operational availability of the IHAWK missile battery or a missile engagement assignment

hawk_msl
IHAWK missile availability; identifies the total number of missiles available for firing

sting_msl
stinger missile availability

tar_tn
target track number; a manual or automatic computer assigned target identification code

tar_bear
target bearing in degrees relative to system center/ vital area

tar_range
target range in miles relative to system center/ vital area

tar_size
target size; individual target track may include 1 to 12 aircraft

tar_speed
target speed in knots

tar_class
target classification; determined through fufillment of identification criteria to include authenticated communications, IFF modes and codes, stereo routes and corridors; valid classifications in this model include: friend, assumed friend, unknown, assumed enemy, and hostile

tar_move
target movement relative to system center/vital area

fit_stat
fighter status; degree of total system operability

fit_size
fighter size; individual fighter track may include 1 to 3 aircraft

fit_loc
fighter location; valid entries include: on runway, enroute to CAP (holding station), 'A' CAP, 'B' CAP, 'C' CAP, and assigned to target

fit_msl
fighter missile; identifies the number of missiles being carried by designated fighter aircraft

fit_range_tar
fighter range to target; identifies the range to a designated assigned target

fit_assign_tn
fighters assigned target's track number

airborne_tar
airborne targets; identifies the derived number of airborne targets in the area of responsibility

airborne_fit
airborne fighter; identifies the derived number of airborne fighters in the area of responsibility

prob_intercept
probability of fighter aircraft completing an intercept on a target

neutr_action
neutralizing action conducted against target/s

APPENDIX H
GLOSSARY

antecedent
The collection of goals in a rule that describe the conditions that must be true in order for the conclusion of the rule to be true.

concept
An idea or abstraction drawn from the specific. The basic unit of description of the items in the taxonomy. A concept can have characteristics or attributes which are represented by properties and roles.

concept definition
The description of a concept in the taxonomy file.

consequent
The conclusion of a rule.

consultation
A single run of the expert system.

control options
Optional features which allow the customizing of an expert system.

definition
A concept for which there is a set of sufficient conditions which define the concept.

expert system
A computer program which addresses problems for which a programming algorithm cannot easily be defined. The air of an expert system is to generate the same solution to a problem that a human would generate faced with the same problem.

goal
A hypothesis, constructed from terms defined in a taxonomy, that is used in a rule. A condition that can be proved either true or not true.

inexact reasoning
Reasoning involving information of less than absolute certainty.

inference
A conclusion based on a premise.

inference engine
The part of a rule-based system that selects and executes rules.

instance
Created during a consultation, an 'instance' is an occurrance of a concept.

knowledge base
A collection of definitions and rules for use with an expert system.

knowledge engineering
The activities of software engineers who acquire knowledge for knowledge-based systems and decide how to represent it for use in the system.

primitives
A concept for which guidelines exist, but for which an absolute

definition cannot be given.

properties

A quality of a concept.

role

A component of a concept. A role includes a value restriction and a number restriction. The value restriction of a role is another concept.

rule base

A collection of if-then rules that describe the interaction of the knowledge defined in the taxonomy.

subsumption relationship

A relationship of concepts whereby the properties and roles of one concept can be applied to another concept. The subsumed concept is a more specific subset of the subsuming concept.

symbol

A character string which stands for or suggests something by reason of relationship, association, convention, or accidental resemblance.

system

A group of interrelated elements, including ideas, principles, rules, and/or procedures, which form a collective entity.

taxonomy

A structured representation of the general information used in an expert system. The taxonomy structure incorporates subsumption relationships for allowing shared knowledge among the terms defined in the taxonomy.

type declaration

A statement placed at the beginning of a taxonomy file which is used to declare a property or role used in the taxonomy.

LIST OF REFERENCES

1. Schumaker, Randall P. and Franklin, Jude, "Artificial Intelligence in Military Applications," *Signal*, June 1986.

2. Simpson, Jr., Robert L., "Applications of AI Capabilities," *Signal*, August 1986.

3. Boden, Margaret, *Artificial Intelligence and Natural Man*, Basic Books Inc., New York, 1977.

4. Winston, Patrick H., *Artificial Intelligence*, (2nd ed.), Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

5. Scown, Susan, J., *The Artificial Intelligence Experience: An Introduction*, Digital Equipment Corporation, 1985.

6. Davis, Randall and Lenat, Douglas B., *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill International Book Company, New York, 1982.

7. Hayes-Roth, F., Waterman, Donald A., and Lenat, Douglas B., eds., *Building Expert Systems*, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1983.

8. Hayes-Roth, Frederick, "Knowledge Based Expert Systems," *Computer*, October 1984.

9. Fox, A., ed., *Expert Systems State of the Art Report*, Published by Pergamon InfoTech Limited, Maidenhead, Berckshire, England, 1984.

10. FMFM 0-3, *Doctrinal Publications Guide*, U.S. Government Printing Office, Washington, D.C., July 1980.

11. FMFM 5-5, *Antiair Warfare*, U.S. Government Printing Office, Washington, D.C., July 1980.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Gary K. Poock, Code 55Pk Naval Postgraduate School Monterey, CA 93943-5000	1
4.	CAPT. Michael J. Molidor, USMC Rt. 2 Box 321 Frisco, TX 75034	2
5.	CDR. J. Stewart, Code 55ST Naval Postgraduate School Monterey, CA 93943-5000	2
6.	C3 Academic Group, Code 74 Prof. M. K. Sovereign Naval Postgraduate School Monterey, CA 93943-5000	2
7.	Naval War College Wargaming Department Sims Hall, C109 Newport, RI 02841-5010	1
8.	MAJOR T. J. Brown, USAF Code 39 Naval Postgraduate School Monterey, CA 93943-5000	1
9.	Capt. Stephen P. Dodd, USMC 10334 Londonderry Rd Houston, Tx 77043	3
10.	T.R. Sivasankaran Admin Sciences 54SJ Naval Postgraduate School Monterey, Ca 93943-5000	3