NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

AN EXPERT SYSTEM
FOR
AVIATION SQUADRON FLIGHT SCHEDULING

by

John B. O'Connor

September, 1991

Thesis Advisor: Martin J. McCaffrey

Approved for public release; distribution is unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
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)* 55	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10 SOURCE OF FUNDING NUMBERS Program Element No · Project No · Task No. · Work Unit Accession Number

11. TITLE *(Include Security Classification)*
AN EXPERT SYSTEM FOR AVIATION SQUADRON FLIGHT SCHEDULING (UNCLASSIFIED)

12 PERSONAL AUTHOR(S) OCONNOR, JOHN B.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	From To	1991, SEPTEMBER	105

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

17. COSATI CODES		
FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
EXPERT SYSTEMS, FLIGHT SCHEDULING

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

An aviation squadron's flight schedule has a major impact on that organization's performance and morale. The ability to consistently draft a correct flight schedule that accounts for all applicable factors, requires a flight schedules officer with significant experience and good judgement. Even with those qualities, the task will normally be a lengthy one. The traditional procedure of using grease boards is antiquated in this age of microcomputers and user friendly software. An integrated database application and expert system would provide the capability of expediting the flight scheduling process while simultaneously producing a consistently high quality schedule. It would also provide the training to elevate non-expert schedulers to an expert level of performance.

20. DISTRIBUTION/AVAILABILITY ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL MARTIN J. McCAFFREY	22b TELEPHONE *(Include Area code)* (408) 646-2488 · 22c OFFICE SYMBOL AS/MF

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

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Approved for public release; distribution is unlimited.

An Expert System
for
Aviation Squadron Flight Scheduling

by

John B. O'Connor
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1980

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Author: John B. O'Connor
John B. O'Connor

Approved by: Martin J. McCaffrey
Martin J. McCaffrey, Thesis Advisor

William J. Haga
William J. Haga, Second Reader

David R. Whipple, Chairman
Department of Administrative Sciences

ABSTRACT

An aviation squadron's flight schedule has a major impact on that organization's performance and morale. The ability to consistently draft a correct flight schedule that accounts for all applicable factors, requires a flight schedules officer with significant experience and good judgement. Even with those qualities, the task will normally be a lengthy one. The traditional procedure of using grease boards is antiquated in this age of microcomputers and user-friendly software. An integrated database application and expert system would provide the capability of expediting the flight scheduling process while simultaneously producing a consistently high quality schedule. It would also provide the training to elevate non-expert schedulers to an expert level of performance.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Scheduling the flights of aircraft has been a requirement almost since their inception. The shortage of resources (men, money, and materials), and the desire to establish order (and avoid chaos) made flight scheduling a necessity. Ironically, however, it has not benefited from the technological advances that have been so prevalent in the aircraft industry. Most aviation organizations today prepare their flight schedules using identical procedures and resources (paper, pencil, and scheduler knowledge/intuition) that their ancestors worked with 80 years ago.

The introduction of computers to business over thirty years ago was limited to large organizations due to the computer's size and cost. The recent development and proliferation of relatively inexpensive personal computers (PCs) with significant computational power and data storage capacity, has given smaller organizations the opportunity to utilize this technology. Personal computers (PCs) have the potential of notably improving efficiency. In many cases, however, their use has been limited by a lack of awareness of potential applications and inadequate training.

Flight scheduling requires the scheduler to make optimization decisions about the available resources. An accurate accounting of those resources is essential. Traditionally, that accounting has been done manually with pencil, paper, and grease boards. The typically overwhelming amount of information has resulted in errors by the manual system.

Database management systems (DBMS) provide a means to enter, store, edit, sort, and report large amounts of data. Relational databases can be designed to represent real world entities. Using those designs, specific database applications can be constructed that are tailored to a organization's needs. The information required by the aviation flight scheduler is ideally suited for such a relational database application.

Even with perfectly accurate resource information, the schedulers must abide by numerous regulations, policies, and guidance from senior officials when making the scheduling optimization decisions. That requires broad experience and good judgement by the scheduler that uses a manual system.

Computer based expert systems have been used worldwide by many organizations (Feigenbaum, McCorduck, and Nii, 1988). The goals of such systems are to improve the organization's efficiency (by reducing task completion time), and to increase standardization of decisions that must be made to complete the system's tasks. These goals are achieved by capturing the knowledge of a recognized expert in the system's field, and then representing that knowledge in a computer program. The expert system computer program can then be used by non-experts to guide and assist them in their required system tasks. This technology could be aptly applied to aviation flight scheduling. The knowledge of the expert could guide the non-expert flight scheduler in considering essential factors, constraints, and applicable policies and regulations when making the scheduling decisions.

This thesis will discuss the use of a computer based expert system for aviation squadron flight scheduling. The specific requirements analysis will be for a United States Navy Light Airborne Multi-Purpose (LAMPS) MK III helicopter squadron. It will address the following research questions:

- What is the knowledge used in flight scheduling?
- What are the required system databases, and what is the optimal way to integrate them?
- What models will accurately represent the "expert", and how should they be constructed?
- What are the requirements to implement such an expert system in an aviation squadron?

Chapter II will provide an overview of aviation squadron flight scheduling. Specific attention will be directed toward the system data flow, and knowledge required by the flight scheduler.

Chapter III will be a summary of expert systems. It will discuss their technical concepts, provide definitions for necessary terms, and analyze their application in historical, present, and future context.

Chapter IV will discuss the required system databases. Each of the databases identified in the system analysis will be outlined. The relationships between the databases, and the procedures to integrate them will be summarized.

Chapter V will analyze the models required to represent the expert aviation flight scheduler. Specific references used in the formation of the models will be applicable for the LAMPS MK III community.

Chapter VI will present the results achieved in building a system prototype using Ashton-Tate's dBase IV (DBMS) and Paperback Software's VP-Expert (expert system shell).

Finally, Chapter VI will provide the conclusions and recommendations for use of a computer based expert system to improve aviation squadron flight scheduling.

II. FLIGHT SCHEDULING

A. OVERVIEW

A flight schedule is an organization's plan to accomplish specific missions with its available resources. It details the mission tasking, assigns the required squadron aircrew, specifies the aircrew briefing time and the event starting and ending times, assigns a platform to accomplish the mission, and provides any additional details required to successfully complete the mission. A squadron will typically write a flight schedule for every 24 hour period, and will occasionally write a weekly flight schedule for long range planning purposes. The flight schedule is approved and signed by the squadron Commanding Officer. Once signed, it is considered an official order to be followed by squadron personnel. It provides a means for orderly allocation of personnel and material, which helps ensure that those resources are available when needed. This chapter will provide an introduction to a flight schedule's information requirements and the effort and events required to write it.

The operations department is responsible for the daily production of the flight schedule in a Navy aviation squadron. The flight schedule is a direct reflection on a squadron's ability to carry out its assigned missions and obligations.

If for example, a squadron had very few scheduled flights for several days, that might be an indication that the maintenance department was unable to keep the squadron aircraft in an airworthy state. The schedule can also build, or erode squadron morale. The efficient scheduling of men, money, and material which accomplishes missions with a minimum of confusion, creates squadron esprit-de-corps and reinforces the subordinate's confidence in the organization. The constant failure to complete scheduled missions, or the need to reassign resources due to schedule errors and omissions, does just the opposite. An individual (usually a Navy LTJG or LT) within the operations department is assigned as the squadron flight scheduler. The billet's complexity and importance demands that its holder possess broad knowledge and experience of the squadron's operations. That intelligence is normally acquired from being attached to the squadron for at least a year, and successfully completing an overseas deployment.

B. INFORMATION REQUIREMENTS FOR FLIGHT SCHEDULING

Successful flight scheduling (like nearly all decision making activities) depends upon access to accurate data, knowledge of regulations, experience, and intuition. The sources of data and regulations are widely dispersed, which increases the task's complexity.

1. Aircraft Availability

The squadron's maintenance department and detachments are responsible for providing the flight scheduler with accurate data on aircraft availability. The information must include whether the aircraft is available for flying on a specific date. It is also important to know the hour that it can first be flown to minimize conflicts with any maintenance that might have to be performed prior to flight.

Even an aircraft that is flyable, however, might not be able to accomplish the assigned mission if the necessary equipment is not functional. To preclude that possibility, the flight scheduler and the maintenance department use standardized codes to describe any flight restrictions for an aircraft. Examples of these codes are aircraft limited to day flight only, visual flight rules only, no shipboard use, or requiring a post maintenance check flight. All of those restrictions would require special attention by the flight scheduler to abide by the appropriate scheduling regulations.

The aircraft's maintainers (maintenance department or detachment) will also inform the scheduler of the maximum number of hours that the aircraft can be flown prior to its required preventative maintenance.

The final consideration for aircraft availability concerns the financial resources. A squadron is typically allocated a certain amount of money to spend on aviation fuel during each fiscal quarter. The squadron's maintenance and operations departments will closely monitor the fuel budget. That can result in aircraft non-availability even though there are no maintenance restrictions.

2. Trainer Availability

The squadron's operations department is responsible for determining the availability of trainers required for flight qualifications. That information is used by the flight schedules officer to augment the available aircraft in meeting the squadron's mission and training requirements.

In the LAMPS MK III community, the flight trainers consist of a few multi-million dollar static and dynamic simulators. Those few trainers (divided equally between the U.S. east and west coasts), must be fairly apportioned among all the squadrons and hundreds of pilots and aircrewmen. To achieve that goal, they are centrally controlled. On the west coast, the community's Fleet Replacement Squadron (FRS) is the central point that allocates trainer time to each squadron. They notify the squadrons of the dates, times, type, and number of the trainer for which they are scheduled. Each squadron is then responsible for scheduling missions and crews to efficiently utilize that allocated time.

3. Missions

Each squadron receives both formal and informal mission tasking. The west coast LAMPS MK III community normally receives its formal mission tasking from the

Commander Anti-Submarine Warfare Wing U.S. Pacific Fleet (COMASWWINGPAC). The majority of that tasking is received in a monthly meeting that is attended by each squadron's operations officer. That tasking can include a variety of missions such as providing Deck Landing Qualification (DLQ) training for ships and aircrew, Landing Signal Enlisted (LSE) training, data link training, weapons range training, and logistics transfers. Additional formal tasking can be received by the operations department at any other time via phone calls or other meetings with the squadron's superiors.

Informal mission tasking is comprised of the squadron's internally generated missions, and those missions which the squadron accepts without formal tasking from commands which are outside the squadron's chain of command. A great deal of the informal mission tasking is due to the special nature of the LAMPS community. Unlike the majority of Naval Aviation squadrons which train and deploy as a single unit, the LAMPS squadrons exist to train and deploy numerous detachments to small aviation-capable ships such as cruisers, destroyers, and frigates. The LAMPS squadrons always maintain a core of resources ashore to support their deployed detachments. That is in direct contrast to the aircraft carrier based squadrons which embark all squadron resources.

Once designated by COMASWWINGPAC to support a specific ship with a detachment, the squadron will assign a portion of their pilots, aircrewmen, and maintenance personnel to a newly formed detachment that is an administrative subordinate to the squadron. The detachment is also operationally subordinate to their assigned ship. They coordinate with the ship to determine embarkation periods, and the dates and times of any additional tasking that the ship's Commanding Officer requests. The detachment must then communicate those missions to the squadron's operations department so that they may be scheduled with the appropriate allocation of necessary resources.

The remaining informal mission tasking is generated internally in the squadron. Those missions are in response to squadron training and safety department inputs that inform the operations department when a flight qualification or training requirement needs to be completed or renewed.

4. Flight Training Requirements

The squadron's training and safety departments are responsible for maintaining the database information concerning squadron pilot and aircrew completed flight training and qualifications. There are regulations that direct the flight training and qualification requirements. Some of the primary instructions include the NATOPS General Flight and Operating Instructions (OPNAVINST 3710.7), the SH-60B Naval Aviation Training and Operations Standardization (NATOPS) flight manual, the COMASWWINGPAC Training and Readiness Manual (TREADMAN), the squadron's pilot training syllabus, and the squadron's standard operating procedures (SOP). The training and safety departments monitor the squadron's success at meeting those requirements. They inform the operations department when a mission needs to be scheduled to complete or renew a qualification or training requirement. They also update their database upon completion of that mission.

5. Aircrew Availability

The squadron flight scheduler must keep a database that indicates the availability of aircrew on a specific date and time. In this context, aircrew is a term that is being used to describe both squadron pilots and aircrewmen (airborne mission system operators and search and rescue (SAR) crewmen). A snivel log is used to provide the required information. The snivel log is a time honored Navy tradition that usually means a standard notebook which aircrew use to record the dates, times, and reasons that they desire to be unavailable for scheduling. A snivel can be made for a multitude of reasons. Examples of snivels are school attendance, leave (vacation), personal reasons, etc. The flight scheduler normally respects the snivel, but may choose to disregard it if the squadron's mission commitments are more important.

C. FLIGHT SCHEDULING PROCEDURES

The operations department flight scheduler must use the available information on the mission requirements, and aircraft, trainer, and aircrew availability to formulate the flight schedule. It basically is a plan to optimize the squadron's resources in meeting its assigned tasks.

The starting point is normally the determination of the missions required. Those missions are then prioritized by the flight scheduler. The priorities may be based on the scheduler's experience, or guidance that the scheduler has received from his/her superiors. Typical examples of such priorities would be a DLQ period having priority over a squadron training flight, or a weapons range period having priority over an observer's familiarization flight. The prioritized mission requirements are then compared to the available aircraft and trainer resources. If there are insufficient platforms to accommodate all missions, then the accuracy of the mission prioritization becomes even more important since the lower priority missions will not be scheduled. If there are more platforms than required, then the scheduler will normally create additional training missions to be scheduled, subject to budgetary constraints. The available aircrew are then compared with the list of missions to be scheduled. The list of available aircrew are subdivided according to the flight qualifications that they have achieved. Examples of those include helicopter aircraft commander (HAC), helicopter second pilot (H2P), functional check pilot (FCP), NATOPS instructor, instrument check pilot, etc. The high priority missions are the first to be assigned platforms and aircrew. The flight scheduler will continue this method to schedule the remaining missions. Many of the decisions that the flight scheduler makes requires experience, good judgement, and intuition in order to arrive at the optimal plan. Those heuristics fill the void where there is little or no written guidance for the flight scheduler. The mission prioritization discussion was an example of this. The process of first assigning highly qualified aircrew to high priority missions was an additional example of the application of flight scheduling heuristics.

A summary of the flight scheduling procedures and information requirements that were discussed in the last two sections, can be visually shown in a data flow diagram (DFD). A data flow diagram is a *graphic tool for depicting* the partitioning of a system into a network of activities and their interfaces, together with the origins, destinations, and stores of data (Page-Jones, 1988, p.351). They are one of the primary structured analysis tools, and are used to assist in defining a system's requirements and to gain a better understanding of an existing system. Figure 1 is the overall data flow diagram depicting the LAMPS MK III squadron flight scheduling process.

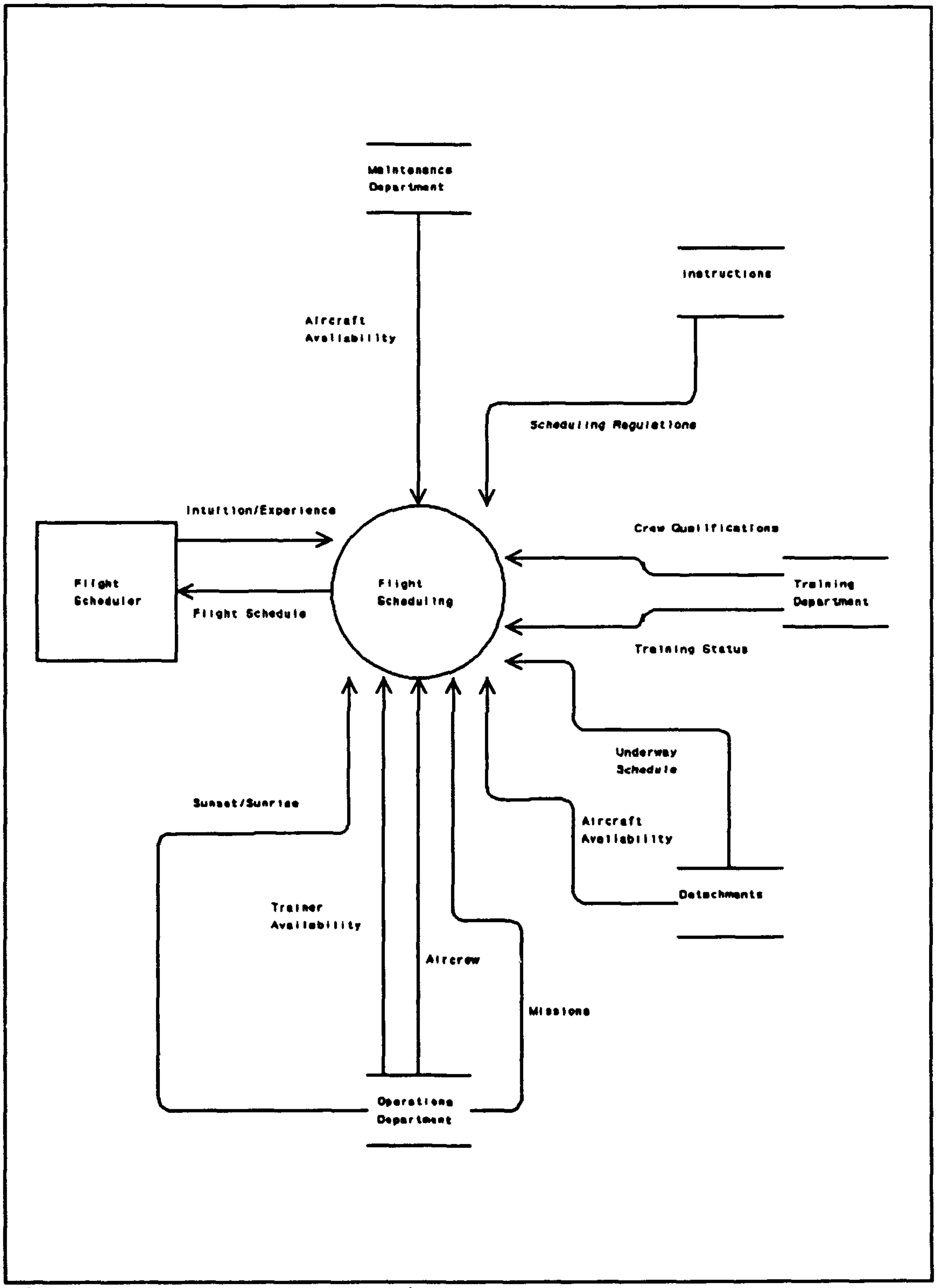


Figure 1. Flight Scheduling Process Context DFD

III. EXPERT SYSTEMS

A. OVERVIEW

Expert systems have been used successfully by many professions during the last ten years. They have proven to be practical means of improving the decision makers productivity, increasing the consistency of the decisions, and providing training and support for users that are non-experts in the domain. Those advantages indicate that expert systems offer potential for improving the flight scheduling process. To fully appreciate that potential, it is necessary to understand what an expert system is. This chapter will introduce expert systems and the terms commonly associated with them, discuss their evolutionary history, present the typical expert system architecture, review the process of knowledge acquisition and expert system development, list the benefits and problems of expert systems, discuss types of tools which are available to build expert systems, and conclude with what the future holds for them.

1. Definition of Expert Systems

An expert system is a computer program that simulates human reasoning in solving a specific domain problem, or providing advice in an area that would normally require a human expert. Users of expert systems may already be recognized experts in the system's subject area. Those individuals use the expert system as knowledgeable assistants or to improve their productivity. Expert systems may also be utilized by non-experts whose decision making skills can be raised to the expert's level of performance, while they are simultaneously receiving expert training. Expert systems are used to propagate scarce knowledge resources for improved, consistent results. The knowledge of an expert system consists of facts and heuristics. The facts constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in the field. The heuristics are mostly private, little discussed rules of good judgment that characterize expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and the quality of a knowledge base it possesses. (Awad, 1988, p. 358) Ultimately, such systems could function better than any single human expert in making judgements in a specific, usually narrow expertise area (referred to as a domain) (Turban, 1990, p. 424).

There are several characteristics that identify an expert system and differentiate them from more conventional application programs. It has extensive specific knowledge from the domain expert, which comes from years of experience at the task. That knowledge allows the expert system to simulate human reasoning about a problem domain, rather than simulating the domain itself. The expert system reasoning is accomplished through symbol manipulation. It arrives at its conclusions and answers through the use of search techniques rather than a sequential algorithmic method. Those searches can be accomplished by either forward chaining (searching for a goal given certain conditions), or backward chaining (determining what conditions must exist for a certain goal to be achieved). It also provides support for solving problems by heuristic or approximate methods which are not guaranteed to succeed. Advanced forms of expert systems have a limited capacity to infer new knowledge from existing knowledge or conditions. Finally, an expert system has the capability to explain to the user the reasoning it used in arriving at a certain conclusion or decision. (Jackson, 1990, p. 4)

2. History of Expert Systems

Expert systems are an outgrowth of the computer research conducted in artificial intelligence (AI). That research has been ongoing since the late 1950's. AI may be divided into several independent categories (Awad, 1988, p. 357):

- Natural Language
- Robotics
- Expert Systems
- Neural Networks
- Cased Based Reasoning

The use of natural language for computers involves software capable of reading, speaking, and understanding the human language. There has been very limited success in this area, but it is often stated that the progress made in expert systems would not have been possible without the extensive natural language research (Rolston, 1988, p. 3). Robotics is the source of the smart robots that have been developed for industry (such as the automotive industry) to augment and replace mundane, repetitive human tasks. Expert systems began to emerge as a separate research area as early as the middle 1960's. Early expert systems were more academic in nature, such as chess games. Continued research led to practical applications such as MYCIN, which proved to be a successful medical diagnosis aide. By the middle 1970's, several expert systems had begun to emerge. (Rolston, 1988, p. 3) Today, expert systems are used in a variety of environments and professions. For instance, they are being used by American Express Corporation to bring consistency and control over decisions to grant customer credit, by Japanese steel companies to maintain high quality production despite a lack of human experts, throughout DuPont Corporation to meet end-user computing needs and gain competitive industry advantage, and by personal computer users in the United States that are using programs such as Andrew Tobias' Tax Cut software to quickly and correctly complete myriad income tax forms. (Feigenbaum, McCorduck and Nii, 1989)

3. Architecture of Expert Systems

The expert systems that are in existence are not all alike. That should be expected due to the wide variety of uses. It is possible, however, to list common components that comprise a typical expert system architecture. The components include the system user, the user interface, the inference engine, the knowledge base, the explanation facility, a knowledge refining or update facility, the knowledge engineer, and the expert. They are shown in Figure 2 (Turban, 1990, p. 431), along with their relationships.

The user interface is the way the user communicates with the system. It becomes the user's external view of the system and should be as user friendly as possible. That is especially true for systems designed to be used by inexperienced users. Experience has shown that this is best accomplished through the use of graphics, menus, simple pointing devices such as a mouse, and similarities to the user's natural language.

The knowledge base is the memory of the expert system. It must contain all the information the expert uses in making his/her decisions. An expert system's performance is directly related to the percentage of the expert's knowledge expressed in the knowledge base. It is comprised of both facts and heuristics. Factual knowledge is that which is commonly accepted as empirical truths by experts in the domain field.

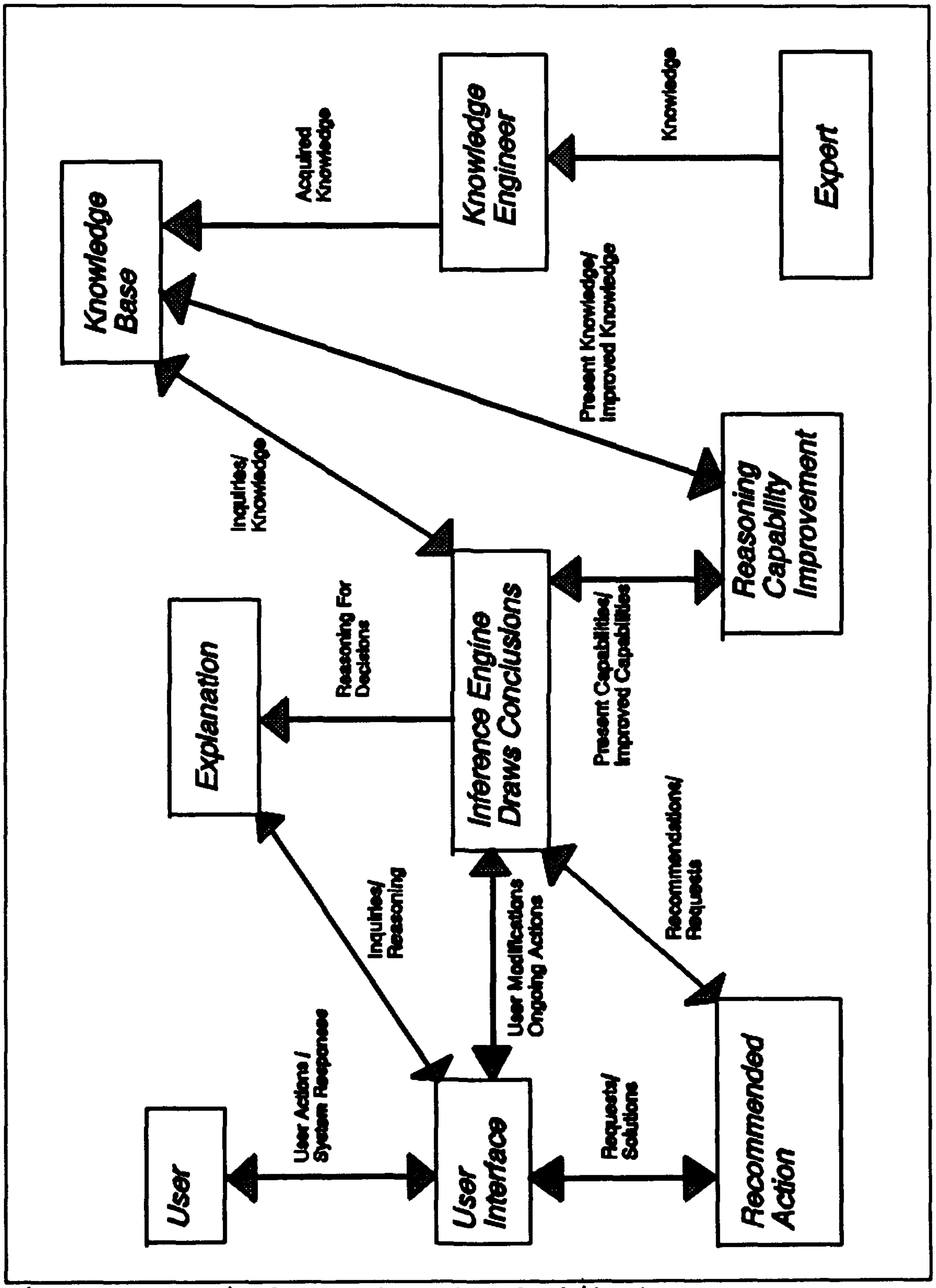


Figure 2. Typical Expert System Architecture

Heuristics, however, are more like rules of thumb that the expert uses. They are based on the expert's experience in the domain field. The facts and heuristic knowledge are combined in a knowledge representation scheme. One common method of knowledge representation involves the use of rules that are comprised of if/then statements. Those rules can then be chained together to simulate the line of reasoning that the expert would make in solving a problem or arriving at a decision.

The inference engine is the brain of the expert system. It contains the inference strategies and controls for manipulating the facts and the rules. The major elements of the inference engine are (Turban, 1990, p. 433):

- An interpreter (rule interpreter in most systems), which executes the chosen agenda items by applying the corresponding knowledge base rules.
- A scheduler, which maintains control over the agenda. It estimates the effects of applying inference rules in light of item priorities or other criteria on the agenda.
- A consistency enforcer, which attempts to maintain a consistent representation of the emerging solution.

The explanation facility is designed to provide the user with the ability to review the reasoning the system used in determining its conclusion. This is an important function for increasing the user's confidence in the system, and for the training of the non-expert user. That transfer of expertise was previously mentioned as one of the prime advantages of expert systems. The explanation facility should

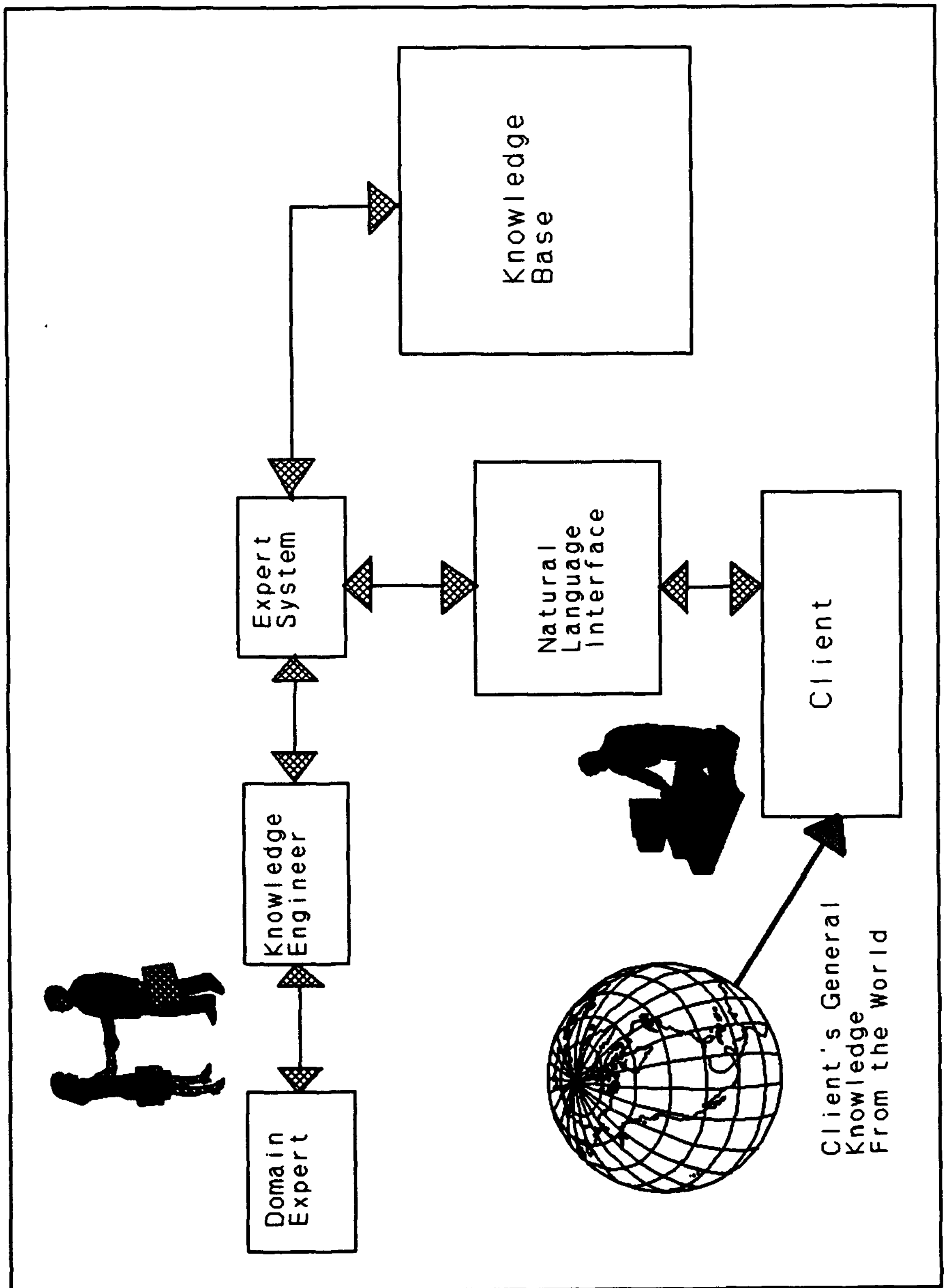


Figure 3. Human Interactions with Expert Systems

be capable of explaining the expert system behavior by interactively answering questions such as:

- Why was a certain question asked by the expert system?
- How was a certain conclusion reached?
- Why was a certain alternative rejected?
- What remains to be established before a final diagnosis can be determined? (Turban, 1990, pp. 432-433)

The knowledge refining or update facility addresses the reality that knowledge is not static. The expert system must continue to expand its knowledge base accordingly for it to remain an effective tool for the user. The refinement or update can be accomplished via any one of three basic methods. The simplest and most commonly used method is done manually by a knowledge engineer who interprets what the domain expert says. This knowledge transfer will be discussed in greater detail in the next section. The second method involves the domain expert refining the knowledge base directly. This is the current state of the art in expert systems. The third and final method requires the system to learn from itself. This is one of the elusive goals of artificial intelligence research. Software with such a feature would have the capability to learn from its experience without the necessity for manual human intervention. That process is still in a conceptual state, and is the subject of a great deal of AI research. (Rolston, 1988, p. 10) Figure 3 (Awad, 1988, p. 363) is a depiction of the expert system human interactions.

4. Knowledge Acquisition and Representation

The knowledge acquisition process provides the means for building the knowledge base. It involves the interaction between the domain expert and the knowledge engineer. The transfer is usually accomplished by a series of lengthy and intensive interviews between a knowledge engineer, who is normally a computer specialist, and a domain expert who is able to articulate his/her expertise to some degree. It is estimated that this form of labor produces between two and five units of knowledge (rules of thumb) per day. That rather low output has led researchers to look upon knowledge acquisition as the bottleneck of expert systems applications. There are a number of reasons why productivity is typically so poor. Some of those reasons include (Jackson, 1990, pp. 7-8):

- Specialist fields have their own jargon, and it is often difficult for experts to communicate their knowledge in everyday language.
- The facts and principles underlying many domains of interest cannot be characterized precisely in terms of a mathematical theory or a deterministic model whose properties are well understood.
- Experts knowledge includes much more than mere facts or principles. The heuristic rules are the most difficult for the knowledge engineer to document.
- Human expertise, even in a relatively narrow domain, is often set in a broader context that involves a good deal of commonsense knowledge about the everyday world.

There are many sources of knowledge that provide guidance to the expert. These can be divided into 3 broad areas (Rolston, 1988, p. 5):

- Facts: Statements that relate some element of truth regarding the subject domain.
- Procedural Rules: Well-defined, invariant rules that describe fundamental sequences of events and relations relative to the domain.
- Heuristic Rules: General rules in the form of hunches or rules of thumb that suggest procedures to be followed when invariant procedural rules are not available.

Those areas can be further categorized into specific subject types that the expert is aware of, and draws from when reaching conclusions and making decisions. Figure 4 (Turban, 1990, p.456) gives a breakdown of the types of knowledge that the knowledge engineer must elicit from the expert and document in the knowledge base.

It is necessary to represent the acquired knowledge with appropriate symbols that can be manipulated and processed by the computer. Popular representation schemes include semantic networks, rules, frames, and logic. A semantic network is a collection of nodes that are linked together to form a net. The net should be representative of the real world situation if the semantic network is accurate. Rules are conditional statements that specify an action to be taken, if a certain condition is true. They are typically expressed in the form of if/then statements. They differ, however, from traditional programming if/then statements in that the rules are relatively independent and will probably be based on heuristics. A frame can be likened to an index card. It associates an object with facts, rules, or values that are

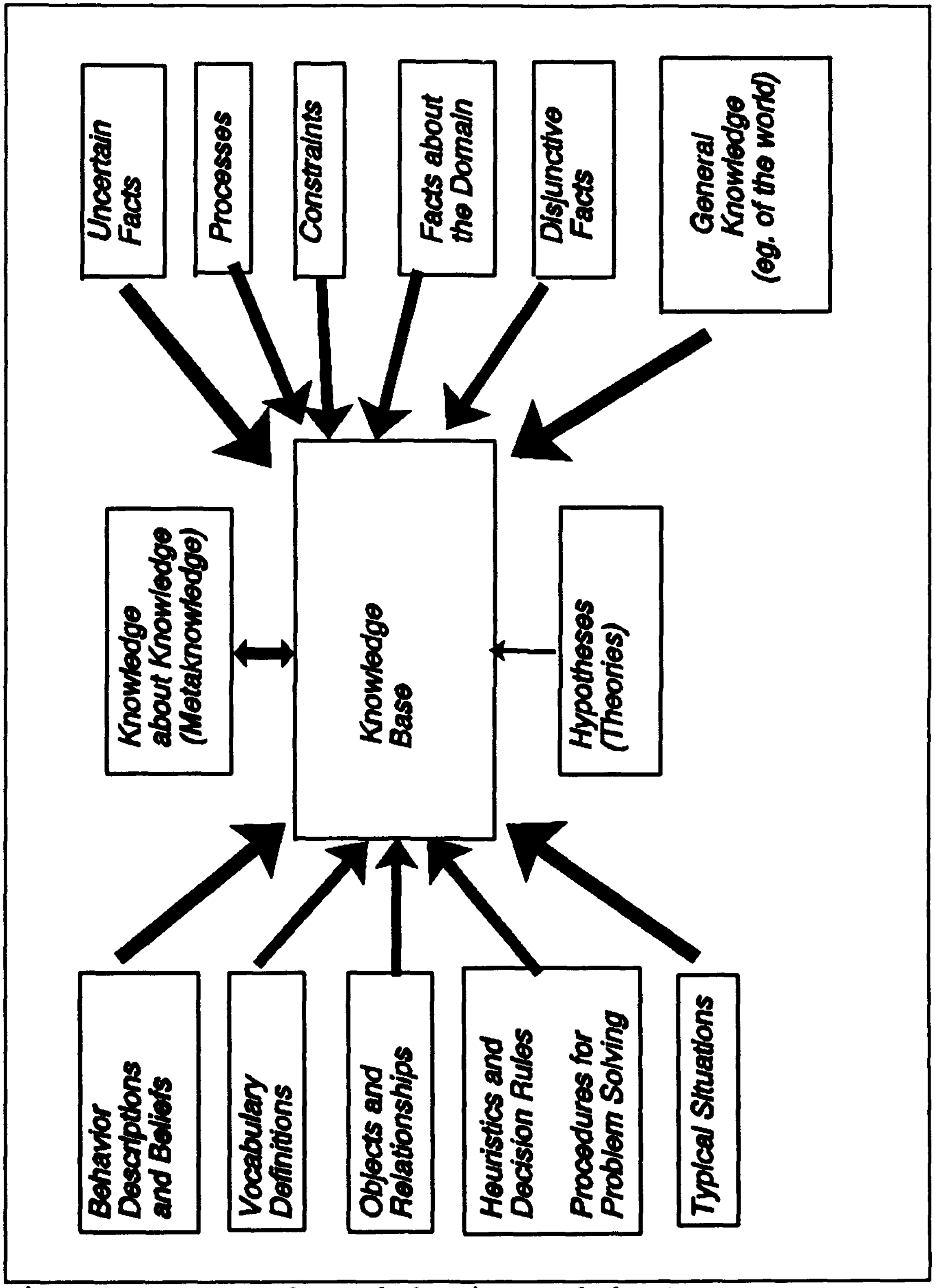


Figure 4. Types of Knowledge in Knowledge Base

stored in a slot. Logic is a system that prescribes rules for manipulating symbols. A widely studied formal language for symbol structures is predicate calculus. A predicate is a statement about an object. The use of logic in forms such as predicate calculus are merely specialized languages for representing knowledge in the form of symbols. (Awad, 1988, pp. 364-366). Expert system developers often use combinations of the above schemes to better represent the knowledge.

Once the acquired knowledge has been encoded, the symbols that represent the knowledge must be tested to ensure their accuracy. Further refinements will probably be necessary to ensure that there is a good representation of the real world situation.

The process of knowledge acquisition and representation can be summarized by showing it as a series of stages such as Figure 5 (Jackson, 1990, p. 221). The feedback between the stages provides the refinement to the solution. That feedback is a characteristic that is prevalent throughout the typical expert system development.

5. Development of Expert Systems

The development of expert systems requires completion of an iterative design process that bears some resemblance to the standard system development life cycle (also known as the waterfall model). The knowledge acquisition process is a key part of the system's requirements analysis. Typical expert

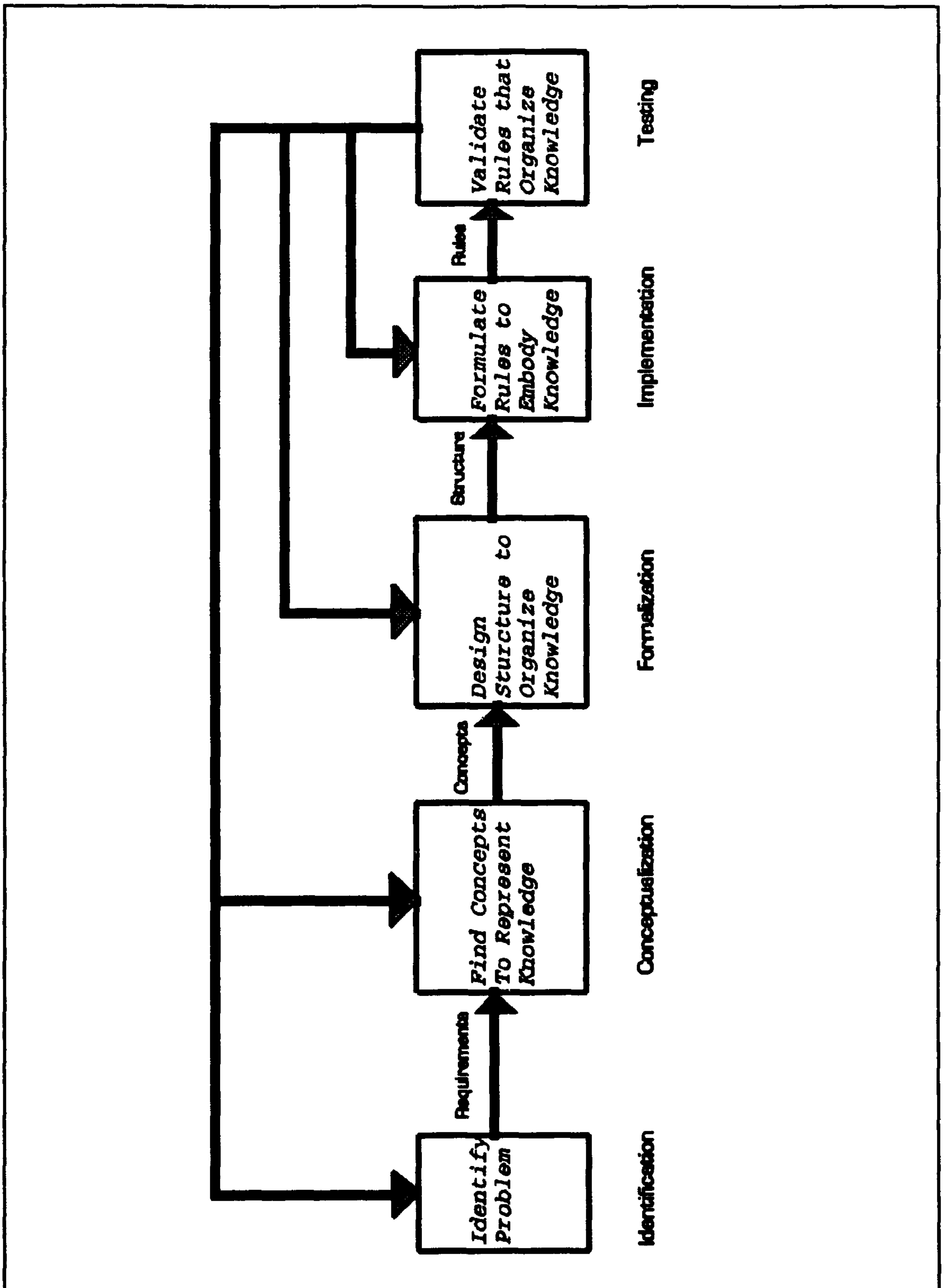


Figure 5. Stages of Acquisition and Representation

systems make extensive use of prototyping during development. Prototyping is an iterative process in which the user evaluates the prototypes and works with the knowledge engineer and programmers to improve the system in incremental steps.

One of the first, and perhaps the most important step in the requirements analysis, is deciding whether the situation that is being evaluated is suitable for an expert system solution. Rolston provides the following guidance:

> "If the problem under consideration can be described in terms of direct definitions and algorithms, it is probably preferable to develop a traditional software solution. If it is ill-defined or requires intensive human judgment (e.g., judging an art contest), it is probably too complex for an expert system." (Rolston, 1988, p. 12)

Using that definition as a guideline, it can be inferred that an expert system would be suitable for a domain that is somewhat structured but which requires the application of human reasoning and inferences about the available domain facts to arrive at a satisfactory conclusion. A suitable expert must also be available to document his/her domain knowledge. Those requirements can be better appreciated by reviewing the general categories of applications that expert systems have been developed for. Those categories include: (Turban, 1990, pp. 436-437)

- Interpretation: Inferring situation descriptions from observations
- Prediction: Inferring likely consequences of given situations
- Diagnosis: Inferring system malfunctions from observations

* Design: Configuring objects under constraints
* Planning: Developing plans to achieve goal(s)
* Monitoring: Comparing observations to plan vulnerabilities, flagging expectations
* Debugging: Prescribing remedies for malfunctions
* Repair: Executing a plan to administer a prescribed remedy
* Instruction: Diagnosing, debugging, and correcting student performance
* Control: Interpreting, predicting, repairing, and monitoring system behaviors

6. Benefits of Expert Systems

There are a great number of benefits associated with expert systems. Well designed systems can be an excellent substitute when there is a shortage of skilled personnel. Even for expert users, the system can act as an assistant to improve their productivity and efficiency. In cases where the knowledge base contains the acquired knowledge of several experts, the expert user will probably benefit and learn from the knowledge of his/her colleagues expressed in the expert system. This tutoring benefit is even more important for the non-expert who can be guided into making the right decisions, and can elevate his/her own skills and knowledge through observation of the system's reasoning. Guiding the non-expert into making the right decision also aids in standardizing the decision making process. (Turban, 1990, pp. 438-440)

7. Limitations of Expert Systems

A major limitation to expert systems development is the bottleneck of knowledge acquisition. The limited number of experts and the difficulty of translating and symbolically representing their heuristics and vocabulary are the major causes of that bottleneck. Other limitations include (Rolston, 1988, pp. 13-14):

- Application must be limited to a specific domain or a small collection of domains
- The application domain must have little need for temporal or spatial reasoning
- The task does not rely on the use of a large body of general or commonsense knowledge

8. Software Tools For Developing Expert Systems

The increasing numbers of expert systems that have been developed and installed in industry during the last ten years is directly related to the improved software tools that are available in the market. That trend has paralleled what has happened in other software areas. The emergence of the personal computer increased the end-user's demand for software that would allow them to meet their own information needs. The development of sophisticated fourth generation software packages/languages, and object oriented programming were the marketplace's responses to that demand.

Early developers of expert systems were dependent on existing programming languages. The data and control structures were typically not suitable to the tasks, which limited the development efforts. A major improvement occurred when an existing expert system named MYCIN was used as the basis for an expert system shell which became known as EMYCIN. The shell is the foundation of the expert system. It typically contains features such a rule language, an indexing scheme for rules, a backward-chaining control structure, an interface between the final consultation program and the end-user, and an interface between the system designer and the evolving consultation program. (Jackson, 1990, pp. 224-225) Expert system shells have given end-users a tool to develop an application for their specific domain, and thereby capture the potential power of expert systems.

There are also special purpose languages that were designed for use with artificial intelligence or symbolic manipulation languages. The most publicized of these include LISP (List Processor), and PROLOG (Programming in Logic). These languages are typically used by more advanced programmers who are building applications that exceed the capabilities of available shells. Figure 6 (Awad, 1988, p. 369) compares the available software tools with their typical users. The end-users are able to use expert system shells and fourth generation languages to develop their application, while the use of AI, third generation, and assembly languages are typically used by more experienced application programmers.

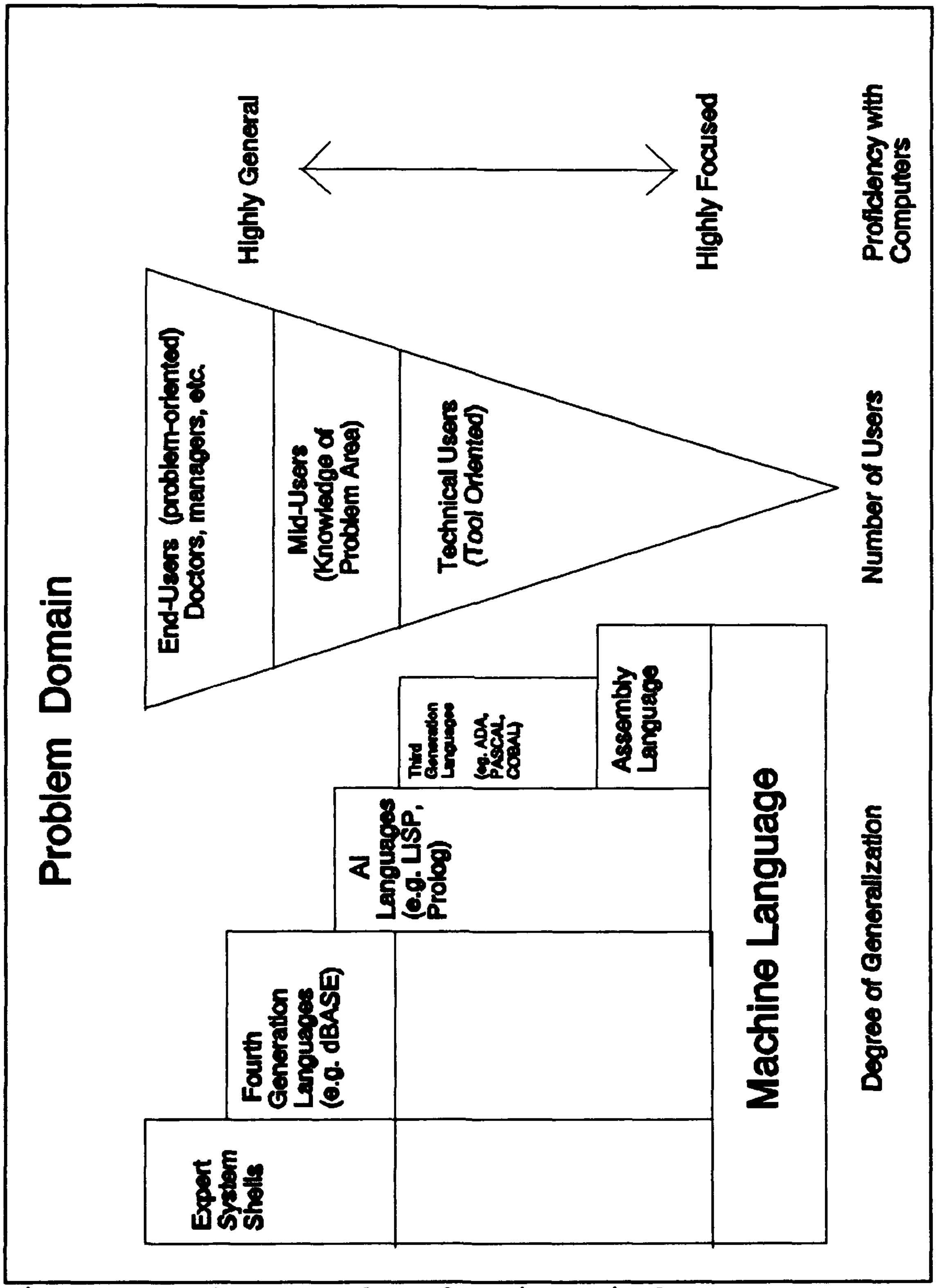


Figure 6. Software Tools and their Typical Users

9. Future of Expert Systems

Expert systems have emerged as a powerful source of information. An indication of their popularity is the fact that there were over fifteen hundred of them in operation in 1987 (Feigenbaum, McCorduck, and Nii, 1989, p. 258) The majority of those systems were developed only a few years earlier due to the increased availability and power of the software development tools. Innovative companies are recognizing that expert systems are having an impact on their business by capturing the knowledge of scarce experts, improving the quality and the consistency of their manager's decisions, providing new revenues from the export of information products, reducing costs due to increased productivity and efficiency, and stimulating innovation among their workers as they consider new ways to solve problems. (Feigenbaum, McCorduck, and Nii, 1989, pp. 260-261)

The advantages of expert systems will become even more important as companies look for ways to reduce costs in recessions, and are forced to cope with a shrinking number of technically qualified workers.

IV. DATABASE INTEGRATION

A. INTRODUCTION

The successful implementation of an expert system for an aviation squadron's flight scheduling requirements will necessitate access by the system to a great deal of squadron data. That data will be used by the expert system's knowledge base to make the proper flight scheduling decisions. It involves ensuring that available pilots are being scheduled for missions that they are qualified for, that the squadron and its detachments are meeting their flight training requirements, that aircraft are being scheduled for missions that they can support with their operational equipment, that applicable regulations are being adhered to, and that all required missions are being scheduled.

The data to support these decisions is typically dispersed throughout the squadron. It is usually recorded and updated with a manual record-keeping system. Examples of these manual systems include the use of grease boards, folders and file cabinets, and logbooks. The process for gathering this information in preparation for writing the flight schedule involves visits, meetings, and phone calls between the flight schedules officer and the individuals in the various departments that are in charge of maintaining the required data. It is not surprising that such a manual system increases the time required to write a flight schedule, and typically results in a higher error rate. The errors can often be traced to missing information, outdated information, or omissions by individuals when manually scanning the voluminous amount of data.

B. REQUIRED SYSTEM DATABASES

The first steps in determining how to improve this situation, is to decide what data the flight scheduler requires, and which squadron departments are responsible for maintaining that data. To accomplish that, the following subsections will review the squadron operations, training, and maintenance departments. In each case, the department's applicable databases will be identified. Each database will be analyzed to decide what fields should be included for data storage. An example of each database is given in Appendix A. The database key and any foreign keys that are necessary will be identified in Appendix A for each database example.

A database key is a group of one or more attributes that uniquely identifies a row in a database. Every relation has at least one key. A foreign key is a key from another database that is included to link the databases. (Dolan and Kroenke, 1988, p. 139)

1. Operations Department Databases

The operations department is responsible for maintaining the majority of the data required for the squadron's flight scheduling. That is appropriate since it is the department that is writing the schedule. The following subsections describe the operations department databases:

a. *Missions*

The missions database must include the date the missions are supposed to be scheduled for, the mission's starting and ending time, the type of mission (ie. DLQ's, Logistics, ASW, etc.), the mission's location, and any other additional information that is required to successfully complete the mission (ie. sonobuoys required, SAR crewman, etc.).

b. *Aircrew*

The aircrew database must include the detachment assignment (if applicable), the individuals name, birthday, designation (ie. pilot or aircrewman), the last date flown, the land time of that last flight, and the date of the individual's last night flight.

c. *Schedule Database*

The schedule database is used to record the dates and times that aircrew snivel as being not available. It must include the snivel's starting date and time and its ending date and time. To minimize redundancy of database structures, this database can also be used to record aircraft and trainer availability dates and times.

d. Detachment Database

The detachment database is used to record what ships each detachment is assigned to.

e. Trainer Database

The trainer database is used to record the trainer device designation (ie. Weapon System Trainer (WST), Weapons Tactics Trainer (WTT), Operational Flight Trainer (OFT), etc.), and its identifying number.

f. Daytime Database

The daytime database records the predicted sunrise and sunset time for each day. Most squadrons utilize paper printouts for this information. Another method of obtaining this information is to use a separate program (so the data does not have to be re-entered).

g. Event Database

The event database is actually an intersectional relationship that is utilized to express the numerous many-to-many relationships that occur between the system databases. It is comprised of the separate events that are listed on the daily or weekly flight schedule. It records the platform that is being used, the mission that is being accomplished, and the aircrew that have been assigned.

2. Training Department Databases

The training department is responsible for overseeing the squadron's flight and ground training programs. They must coordinate with the operations and safety departments to ensure that all required flight related training is accomplished. That involves maintaining data on flight qualifications that squadron aircrew achieve, required schools and proficiency examinations that must remain current, and completion of the squadron's flight training syllabus requirements. The following discussion pertains to the training department's database requirements for a LAMPS MK III squadron in San Diego.

a. Qualifications Database

The qualifications database records the dates that each aircrew completes his flight related qualifications. This includes the date he was designated a pilot qualified in model (PQM), helicopter aircraft commander (HAC), NATOPS instructor, etc.

b. Training Database

The training database records the date that each aircrew completes required flight related ground schools such as water survival. It also includes the dates that flight proficiency checks were last completed (i.e. NATOPS and instrument checks), and the dates that the squadron and wing's flight training syllabus events were last completed.

3. Maintenance Department Databases

The maintenance department is responsible for supporting squadron operations by ensuring that the squadron aircraft are capable of accomplishing their assigned missions, or are in a state of repair to return them to that capability. It must provide the operations department with list of available aircraft for each date. That list must specify any flight restrictions for each aircraft that would impact the types of missions they could be scheduled for. The list must also specify the starting and ending availability times for each aircraft on the respective dates, and the maximum amount of hours each aircraft can be flown before it requires preventive maintenance. The required information is stored in the aircraft and *schedule* databases.

C. DATABASE RELATIONSHIPS

1. Theory

Proper database design is critical for its efficient operation. Without it, there will be a significant amount of data redundancy, inadvertent deletion of data, excessive requirements for entering new data, and difficulties in querying the databases for required information. The previous section presented the data bases that represent objects in the flight scheduling process. This section will discuss the relationships that exist between those databases.

Relationships between databases can be described in one of three ways. They can be:

- One-to-One
- One-to-Many
- Many-to-Many

a. One-to-One Relationships

A one-to-one relationship is the simplest form. An object relationship is one-to-one if Object A contains Object B as a single-valued object property, and either Object B contains Object A as a single-valued object property or Object B does not contain Object A. Simply put, it means that there can be a maximum of only occurrence for an entity in an object. The key of one of the relations must be stored as an attribute of the other in order to link them together. (Dolan and Kroenke, 1988, pp. 169-174)

b. One-to-Many Relationships

A one-to-many relationship occurs when a record of one type is related to potentially many records of another type (Dolan and Kroenke, 1988, pp. 174-178). An example of this is the relationship between a detachment and its aircraft. A detachment may have many aircraft. In that case, the detachment number would appear more than once in the aircraft database entries. The terms parent and child are sometimes applied to records in one-to-many relationships. The parent record is on the one side of the relationship and the child record is on the many side. The key of the parent relation must be stored as an attribute of the child relation.

c. Many-to-Many Relationships

The third and final type of database relationship is the many-to-many. In that relationship, a record of one type corresponds to many records of the second type and a record of the second type corresponds to many records of the first type. An example of this is that a pilot may be scheduled for many missions, while a mission may utilize many pilots. Many-to-many relationships cannot be directly represented in relations as the previous two could. There are physically not enough fields in each database to represent all the occurrences. The solution to the problem is to create a third relation that shows the correspondence of the databases. That third relation is sometimes called an intersection relation. Each record in an intersection relation contains the keys of each of the related records in the other two relations. (Dolan and Kroenke, 1988, pp. 178-183)

2. LAMPS MK III Flight Scheduling Data Base Relationships

The ten databases used for the LAMPS MK III flight scheduling process were analyzed to determine their relationships. Figure 7 is an entity relationship diagram (ERD), which is a depiction of the databases and their relationships. The single arrows indicate a one relationship,

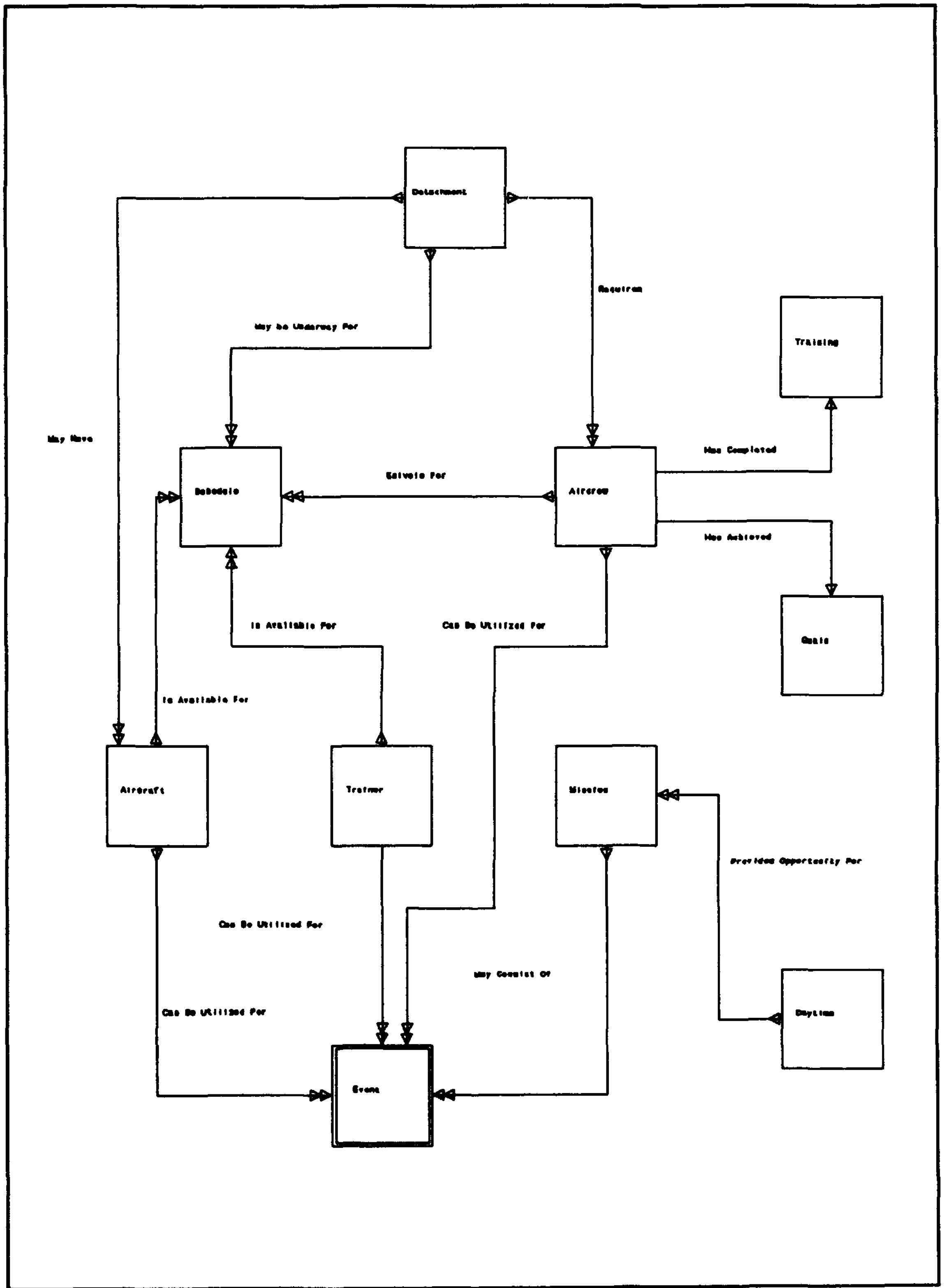


Figure 7. LAMPS MK III Squadron Flight Scheduling ERD

while a double arrow represents a many relationship. The following paragraphs will discuss the depicted relationships.

Detachments have a one-to-many relationship with aircraft since a detachment can simultaneously have many aircraft while an aircraft can only be assigned to one detachment at a time. Detachments also have a one-to-many relationship with schedules since they can have several scheduled underway periods. The last relationship for detachments is a one to many relationship with aircrew. A detachment can have many assigned aircrew, but an aircrew can only be assigned to one detachment at a time.

Aircraft have a one-to-many relationship with schedule since there may be several different periods of time that they may be available to be scheduled. Aircraft also have many-to-many relationships with missions and aircrew. An aircraft can be assigned several missions and a mission may require the use of many aircraft. An aircraft requires many aircrew to fly and aircrew may be assigned to several aircraft for different missions. Event is the intersection database to be used to reflect these many-to-many database relationships.

Trainer has a one-to-many relationship with schedule since it may have several time periods that it is available to be scheduled. It also has many-to-many relationships with aircrew and missions. A trainer can be scheduled for several missions and a mission may require the use of several trainers. A trainer may also have several aircrew scheduled to utilize it, while an aircrewman may be assigned to several trainers for different missions. Once again, event is the intersection database that reflects these many-to-many database relationships.

Aircrew have a one-to-many relationship with schedule since there may be many time periods that they request not to be scheduled. There is a one-to-one relationship between aircrew and training, and aircrew and qualifications. An aircrew can only have one training record, and a specific training record can only belong to one aircrew. Likewise, an aircrew can only have one qualification record, and a specific qualification record can only belong to one aircrew. Aircrew's many-to-many relationships have previously been discussed.

The final relationship is daytime's one-to-many relationship with mission. This just shows that a day may have several missions scheduled.

D. DATABASE INTEGRATION AND IMPLEMENTATION

The integration and implementation should be accomplished by a database application specifically programmed for the flight scheduling system requirements that have been introduced. The database application should be based on a microcomputer in the squadron's operations department. Their are numerous relational DBMS in the commercial market that could be used to accomplish this. Examples of these include

Ashton-Tate's dBase IV, and Borland's Paradox. Unfortunately, there is little organized Navy support for such information needs. It is therefore up to units such as the squadrons to use their internal talents, and the relative ease of the fourth generation languages to fulfill those information systems needs.

Once a squadron has implemented such a database application, they will still be faced with the problem of transferring the information from the training and maintenance departments, to the operations department's flight schedules officer. A manual work-around is to carry floppy disks between offices. A long term solution should be the implementation of a computer network (such as a token-ring) that would connect each of the departments and the Commanding Officer and Executive Officer. The Navy is beginning to design networks into new ship constructions. They have also been implemented on a limited basis in some shore installations. The fact that LAMPS squadrons do not deploy (deploying only detachments), should make network installations a very viable option. Since multiple squadrons are housed in a single hangar, it would also be relatively straightforward to interconnect each of those squadrons. The information needs of the lower level Navy organizations should receive higher funding priority than it appears they presently have.

V. FLIGHT SCHEDULE MODELING

A. INTRODUCTION

The knowledge base of the expert system must accurately model the expert's view of the problem domain for it to be effective. To do that, it must incorporate the applicable requirements, regulations, instructions, and heuristics that guide the expert through the system's decision making process. The development of such an accurate model normally requires an iterative process. Prototypes are evaluated by the expert(s) who identifies any model deficiencies. The knowledge engineer is then responsible for improving the knowledge base by refining the model with the identified requirements. That is a critical process since the expert system's effectiveness is directly related to the breadth and accuracy of its knowledge base.

B. FLIGHT SCHEDULING PROCESS

The sequence of actions that the flight schedules officer completes when drafting a flight schedule is relatively consistent. The fourteen steps involved in the flight scheduling process are as follows: (1) Obtain from the maintenance department a list of aircraft that will be available during the time period to be scheduled. That list should identify the aircraft, the total number of hours that may be flown prior to preventive maintenance, any equipment malfunctions that will still be outstanding that may limit mission assignments, the hour that the aircraft will be ready for an aircrew assignment, and whether a post maintenance check flight (PMCF) will be required. (2) Acquire a list of which trainers have been allocated for the squadron's use during the time period to be scheduled. The flight training devices are normally controlled by the community's Fleet Replacement Squadron (FRS). The list must also specify the type of trainer (WST, WTT, or OFT) and the specific trainer number. (3) Complete a listing of missions that need to be scheduled for the subject time period. The mission details must be explicit enough to account for all scheduling details such as time, location and type of mission (i.e. training, logistics, DLQ's, etc.). (4) Determine what pilots and aircrewmen are available for scheduling during the specific time period. That information is determined by reviewing the snivel log and removing those individuals with valid snivels from the list of squadron flight personnel. (5) Compute the current total of flight hours and night flight hours that the squadron has flown during the month, quarter, and fiscal year. Those numbers should be cross-checked with the maintenance department figures. (6) Determine whether the squadron is on track to achieving its month, quarter, and fiscal year flight hour and night flight hour goals. (7) Prioritize the list of missions that need to be scheduled. (8) Assign missions to the available platforms (aircraft or trainer) that are applicable. Assignments should be started at the beginning of the prioritized mission list. (9) Determine if there is any available aircraft or trainer time that remains unscheduled. (10) If there are additional aircraft and/or trainer times that are still unscheduled, and the squadron has sufficient remaining flight hours for the month, quarter, and fiscal year, additional training missions should be added to the scheduled period. (11) Verify the flight qualifications that each pilot and aircrewman have achieved with the training department. (12) Obtain a list from the training department that shows the status of required flight training for the squadron personnel in flight status. The training department should also specify what they consider to be training priorities. (13) Assign available and qualified pilots and aircrew to the missions that are being scheduled. (14) Obtain necessary approval of the completed schedule after making any requested changes, and disseminate as appropriate.

C. REGULATIONS AND GUIDELINES

There are numerous regulations and guidelines that the flight schedules officer must adhere to when preparing the schedule. The following subsections will discuss those that are found in the NATOPS manual, squadron training syllabus, squadron standard operating procedures (SOP), training and readiness manuals (TREADMAN), and OPNAVINST 3710. The specific examples used will be based on a west coast LAMPS MK III squadron perspective. The applicable regulations and guidelines from each reference are listed in Appendix B.

1. NATOPS Flight Manual

The NATOPS flight manual is issued for each type of aircraft in the Navy's inventory. Its purpose is to standardize the training and operations for those aircrew that fly that aircraft. The overall goal is improved safety. To help in achieving that goal, the manual specifies aircrew proficiency and minimum qualifications for aircrew assignments to missions (Naval Air Systems Command, 1987, p. II-5-2). Those requirements are listed in Appendix B.

2. Squadron Pilot Training Syllabus

A squadron's pilot training syllabus is the Commanding Officer's plan to ensure that the aircrew will be properly trained to accomplish all assigned missions. It augments the training regulations that the squadron's superiors promulgated. The guidelines documented in Appendix B are from a west coast LAMPS MK III squadron pilot training syllabus instruction (Helicopter Anti-Submarine Squadron Light Forty Three, 1989, pp. 1-23). They specify the prerequisites that must be completed prior to an aircrew being scheduled for a squadron training mission, and the intended sequence of those missions.

3. Squadron Standard Operating Procedures (SOP)

A squadron's SOP is issued by the Commanding Officer. It provides a quick means for the Commanding Officer to clarify ambiguous information in other aircrew regulations, or impose stricter operating procedures. The specific guidelines in Appendix B are documented in a west coast LAMPS MK III squadron's SOP (Helicopter Anti-Submarine Squadron Light Forty Three, 1991, p. 4). They detail crew rest requirements, and aircrew assignment policies.

4. Training and Readiness Manual

The training and readiness manual is normally issued by the squadron's wing commander. It is applicable for all squadrons that are operationally subordinate to that wing. It details the training requirements that each squadron must schedule for their aircrew. It also specifies the expiration period for a completed training requirement. The training requirements and currency periods in Appendix B are documented in the west coast LAMPS MK III squadron's wing training and readiness manual (Anti-Submarine Warfare Wing Pacific Fleet, 1991, p. III-1-3).

5. OPNAVINST 3710

The NATOPS general flight and operating instructions are the training and operations guidelines issued by the Chief of Naval Operations that are applicable to all Navy aircrew, regardless of the platform that they fly. Appendix B lists those that are applicable to the flight scheduling process. That includes requirements for NATOPS and instrument proficiency checks, flight physicals, and flight safety schools. They are documented in the Navy instruction OPNAVINST 3710 (Office of the Chief of Naval Operations, 1990).

6. Heuristics

There are innumerable heuristics that each flight scheduler uses depending on the situation. Some common ones were noted during this preliminary system requirements analysis (Interview between T. Jara, LCDR, USN, Helicopter Anti-Submarine Squadron Light Forty Three, San Diego, California, and the author, 03 July 1991). Those heuristics are guidelines that are not formally documented in any of the previously introduced references. They are commonly used by proficient squadron flight schedulers, however, since they tend to minimize flight scheduling conflicts and errors. They are listed in Appendix B.

D. SUMMARY

Flight scheduling is a complex process. An expert system that supports that process would only be effective if it properly modeled the scheduler's real world environment. To accomplish that, the model must incorporate the flight scheduling regulations, requirements, and guidelines. It must also document and use the flight scheduler's heuristics that are used to arrive at optimal solutions. The lists of those regulations, requirements, guidelines, and heuristics in Appendix B clearly show the enormity of the task that the flight scheduler must face on a daily basis. Building a knowledge base that models the flight scheduling process is an important step in developing an expert system prototype for microcomputer use.

VI. PROTOTYPE RESULTS

A. OVERVIEW

Initial efforts were made as part of this thesis to translate the aviation squadron flight scheduling expert system's requirements analysis into a working prototype. A substantial amount of work remains to be done in that area. This chapter will review the progress and accomplishments that were made on the database application and expert system prototype.

B. DATABASE APPLICATION

The prototype database application used Ashton-Tate's dBase IV version 1.1. The goal of the prototype was to have a microcomputer based application that was user friendly. It was recognized that frequent squadron job assignment changes necessitated a system that could be operated with minimal user training. That was to be achieved by using menus throughout the application that would be controlled by simple cursor movements, or preferably a mouse pointer.

1. Ashton-Tate's dBase IV

Ashton-Tate's dBase IV was chosen as the application's data base management system (DBMS) because of its availability at the Naval Postgraduate School campus, and the author's familiarity with its operation. Its strengths are in its predefined data structures and its menu driven Control Center which allow the user to easily create and edit databases, queries, forms, reports, labels, and relatively simple applications. More sophisticated applications require the use of its third generation style programming language. The language is powerful but requires a great deal of time for the user to achieve proficiency at using it. The biggest drawback to that DBMS is that it is not truly relational. That problem results in significantly more effort on the part of the application programmer. There is a capability for Structured Query Language (SQL), but it is not integrated with the remainder of the DBMS which means the user must create separate and redundant data structures.

2. Accomplishments

The ten database structures that were discussed in Chapter IV and listed as examples in Appendix A, were created using dBase IV. They included the missions, aircrew, schedule, detachment, trainer, daytime, event, qualifications, training, and aircraft databases.

Eight programs were written in the dBase programming language. Those programs are included in Appendix C. The features that each provides are shown in order in the following list:

- Centers any input character string
- Assigns aircrew numbers

- Assigns mission numbers
- Rebuilds index files for all databases
- Opens data base files and sets relations for pilot and aircraft snivels
- Validates aircrew number for pilot's snivels
- Validates the time that is entered
- Determines aircraft availability based on date and time

User friendly forms for entering and editing data were created. Those forms are described below and examples of each are included in Appendix D:

a. Aircraft Availability Form

This form was intended to be used by the maintenance department to enter and edit information on the availability and status for each of the squadron aircraft.

b. Mission Information Form

This form would allow the operations department to enter pertinent data for each scheduled mission.

c. Pilot Data Form

This form was designed for the operations department to record required data for each squadron pilot.

d. Pilot Qualification Record Data Form

This form would be utilized by the training department to record the dates the squadron pilot's complete their flight qualifications.

e. *Aircrew Snivel Form*

This form would allow the aircrew to enter the dates and times that they request not to be scheduled.

C. EXPERT SYSTEM PROTOTYPE

The goals of the initial flight scheduling expert system prototype were to obtain practical experience with expert system shells, and to begin the iterative process of validating the models (introduced in Chapter V) of the flight scheduling environment. The initial attempt at a prototype for the aviation squadron flight scheduling expert system used Paperback Software's VP-Expert.

1. Paperback Software's VP-Expert

VP-Expert is an expert system shell. That software program was selected because of its compatibility with business applications, its purported user friendliness, and its availability at the Naval Postgraduate School. Its strengths are its features that allow the programmer to quickly develop a customized user interface between the user and the expert system. The capability of writing a single rule and then compiling it and testing it prior to writing the remainder of the program gives a great deal of flexibility to the programmer. The non-procedural aspects of that capability, can be somewhat disorienting to someone that only has experience with procedural programming languages.

2. Accomplishments

The only significant accomplishment that was made on the expert system prototype was the practical experience gained with an expert system shell. The envisioned prototype required a great deal of interaction between the expert system knowledge base, and the databases. That proved to be very cumbersome due to the significant amount of memory that was required for the temporary variables, and the limitation of VP-Expert that prohibits the use of nested programming loops. The code that was written pertains mostly to the user interface. That code is listed in Appendix E.

VII. CONCLUSION

Like all organizations, an aviation squadron must optimize the use of its available resources. The flight schedule is one of the means by which the Commanding Officer strives to achieve that goal. A well written flight schedule will increase the probability that a squadron will complete its assigned missions. It will also minimize the chaos, anger, and extra work that is caused when flight scheduling errors are uncovered. Shoddily written flight schedules reflect poorly on the organization, and more importantly, can impair squadron morale and readiness.

Flight scheduling is a data intensive activity. A LAMPS MK III squadron can typically have over 100 pilots and aircrew. Each of those individuals have different time periods where they will be unavailable for tasking, and specific qualifications and training requirements the must be achieved while simultaneously completing all squadron assigned missions. Aircraft and training platforms are relatively scarce. The platforms that are available may be unable to be used for specific missions due to equipment malfunctions. The flight scheduler must have current data that gives him/her an insight on the exact status of the squadron resources for the period to be scheduled. That data is usually dispersed throughout the squadron's departments.

The flight scheduler must also be knowledgeable about numerous regulations and guidelines that the squadron must adhere to. Even with perfect data and knowledge about the applicable regulations, a flight schedule may still cause problems. The final requirement for successful flight schedules is the application of judgement and heuristics by the flight scheduler.

The current state of flight scheduling in the typical naval aviation squadron involves the assignment of a relatively junior but experienced officer as the flight schedules officer. That officer learns on the job, and schedules by means of a manual system that involves word of mouth data transfer and posting of pertinent information on grease boards. The process is lengthy, with frequent mistakes. Squadron personnel are forced to respond quickly to problems that arise due to the scheduling mistakes. Experience normally will reduce the error rate, but the officer is subsequently transferred to a new assignment, and the process repeats itself.

The proliferation of microcomputers and user-friendly software have given end users significant new options to meeting their information requirements. The flight scheduling process is certainly an area that could take advantage of such technology. The large data storage requirements, with frequent updates, and queries is ideally suited for a microcomputer based database application. It would be a logical first step in improving the flight scheduling process. The networking of squadron computers would be a very beneficial second step in improving the data transfer. The potential third step would be the implementation of an expert system.

Expert systems have benefitted from the tremendous amount of research that has been conducted in the artificial intelligence (AI) field. Their commercial popularity has paralleled the growth of end user computing during the last seven years. They are being developed worldwide by thousands of organizations that span a wide variety of specialty fields. Those organizations are using the expert systems to help meet their information needs while they are confronting the shortage of technically qualified workers, and the need to reduce costs and improve efficiency in their competitive markets. Expert systems have proven their ability to act as an assistant to an established expert with subsequent productivity and efficiency improvements. They have also been excellent tutors that have helped instruct the non-experts while simultaneously raising their capability to perform near the expert's level. The introduction of commercial expert system shells has reduced the time required for an organization to develop an expert system that is tailored to their needs.

An expert system for aviation squadron flight scheduling must possess a knowledge base that includes all pertinent regulations and guidelines that the scheduler must abide by. It must also incorporate a model of the heuristics that the scheduler uses to refine his/her optimization decisions. There should also be a readily accessible link between the expert system's knowledge base and the squadron's data that is needed for flight scheduling. The capability provided by shells to quickly modify the expert system's knowledge base without disturbing the remainder of the program, indicates that it should be possible to develop a generic flight scheduling expert system. The end users could then incorporate that knowledge which was specific to their situation.

To achieve those discussed benefits of database applications, networking, and expert systems in a reasonable period of time, the end user organizations in the Navy such as the aviation squadrons must take the initiative. There is a tremendous amount of untapped talent that could be applied to the tasks by organizations with vision. That principle was clearly demonstrated by Training Squadron Twenty Six (VT-26) which internally developed their own computer network, computer aided scheduling system, and computer aided training (Interview between F. Bosio, CDR, USN, TRARON 26, Beeville, Texas, and the author, 24-25 June 1991). The designation of squadron personnel to act on the organization's information needs would help ensure that those needs get met in a professional manner, and would prevent the occurrence where nothing gets done because its everybody's job. The argument that there are insufficient personnel and that they are needed elsewhere certainly has merit. It can be countered, however, by pointing out that access to proper information in a timely manner has significant direct impacts on an organization's productivity, efficiency, and morale.

APPENDIX A: EXAMPLES OF FLIGHT SCHEDULING DATABASES

Database Legend:

- Database Key = *
- Foreign Key = #

A. Missions Database:

Field	Field Name	Type	Width	Dec	Index
* 1	MISSION_NO	Numeric	10		Y
# 2	DATE	Date	8		Y
3	MSTRT_TIME	Numeric	4		Y
4	MEND_TIME	Numeric	4		Y
5	MSN_TYPE	Character	20		N
6	LOCATION	Character	40		N
7	ADDED_INFO	Character	50		N
8	SCHEDULED	Logical	1		N

B. Aircrew Latabase:

Field	Field Name	Type	Width	Dec	Index
* 1	AIRCREW_NO	Numeric	5		N
# 2	DET_NO	Numeric	2		N
3	LAST_NAME	Character	25		N
4	FIRST_NAME	Character	25		N
5	MI	Character	2		N

6	SSN	Character	11		N
7	BIRTHDAY	Date	8		N
8	DESIG	Character	30		N
9	AVAILABLE	Logical	1		N
10	AVAIL_RSN	Character	30		N
11	LAST_FLOWN	Date	8		N
12	LAST_LNDTM	Numeric	4		N
13	LST_NGTFLT	Date	8		N

C. Schedule Database:

Field	Field Name	Type	Width	Dec	Index
* 1	SSTRT_DATE	Date	8		Y
* 2	SSTRT_TIME	Numeric	4		Y
* 3	SEND_DATE	Date	8		Y
* 4	SEND_TIME	Numeric	4		Y
* # 5	AIRCREW_NO	Numeric	5		Y
* # 6	BUNO	Numeric	6		Y
* # 7	DEVICE_DES	Character	6		Y
* # 8	DEVICE_NUM	Numeric	9		Y
* # 9	DET_NO	Numeric	2		Y

D. Detachment Database:

Field	Field Name	Type	Width	Dec	Index
* 1	DET_NO	Numeric	2		Y
2	SHIP	Character	40		N

E. Trainer Database:

Field	Field Name	Type	Width	Dec	Index
* 1	DEVICE_DES	Character	6		Y
* 2	DEVICE_NUM	Numeric	9		Y

F. Daytime Database:

Field	Field Name	Type	Width	Dec	Index
* 1	DATE	Date	8		Y
2	SUNRISE	Numeric	4		N
3	SUNSET	Numeric	4		N

G. Event Database:

Field	Field Name	Type	Width	Dec	Index
* # 1	DEVICE_DES	Character	6		Y
* # 2	DEVICE_NUM	Numeric	9		Y
* # 3	MISSION_NO	Numeric	10		Y
* # 4	BUNO	Numeric	6		Y
* # 5	AIRCREW_NO	Numeric	5		Y

H. Qualifications Database:

Field	Field Name	Type	Width	Dec	Index
* # 1	AIRCREW_NO	Numeric	5		Y
2	PQM	Date	8		N
3	H2P	Date	8		N

4	ATO	Date	8		N
5	LSO	Date	8		N
6	HAC	Date	8		N
7	FCP	Date	8		N
8	NATOPSINST	Date	8		N
9	INSTR_INST	Date	8		N
10	OFT_INST	Date	8		N
11	WTT_INST	Date	8		N
12	WST_INST	Date	8		N

I. Training Database:

Field	Field Name	Type	Width	Dec	Index
* # 1	AIRCREW_NO	Numeric	5		Y
2	NATOPS_EXP	Date	8		N
3	INSTR_EXP	Date	8		N
4	FT_PHY_EXP	Date	8		N
5	WATER_EXP	Date	8		N
6	AV_PHY_EXP	Date	8		N
7	NIGHT_EXP	Date	8		N
8	SHIP_EXP	Date	8		N
9	SAR_EXP	Date	8		N
10	OFT1	Date	8		N
11	OFT2	Date	8		N
12	OFT3	Date	8		N
13	WST1	Date	8		N
14	WST2	Date	8		N

15	WST3	Date	8	N
16	AC1	Date	8	N
17	AC2	Date	8	N
18	AC3	Date	8	N
19	AC4	Date	8	N
20	AC5	Date	8	N
21	AC6	Date	8	N
22	AC7	Date	8	N
23	AC8	Date	8	N
24	AC9	Date	8	N
25	AC10	Date	8	N
26	AC11	Date	8	N
27	SP1	Date	8	N
28	SP2	Date	8	N
29	SP3	Date	8	N
30	SP4	Date	8	N
31	SP5	Date	8	N
32	ASW1	Date	8	N
33	ASW2	Date	8	N
34	ASW3	Date	8	N
35	ASW4	Date	8	N
36	ASW5	Date	8	N
37	ASW6	Date	8	N
38	ASW7	Date	8	N
39	ASW8	Date	8	N
40	ASW9	Date	8	N

41	ASW10	Date	8	N
42	SURV1	Date	8	N
43	SURV2	Date	8	N
44	SURV3	Date	8	N
45	EW1	Date	8	N
46	EW2	Date	8	N
47	AAW1	Date	8	N
48	CCC1	Date	8	N
49	CCC2	Date	8	N
50	CCC3	Date	8	N
51	SHIP1	Date	8	N
52	SHIP2	Date	8	N
53	SHIP3	Date	8	N
54	SAR1	Date	8	N
55	SAR2	Date	8	N
56	SAR3	Date	8	N
57	FORM1	Date	8	N
58	CGO1	Date	8	N
59	NAV1	Date	8	N
60	NAV2	Date	8	N
61	HET1	Date	8	N
62	HET2	Date	8	N
63	HET3	Date	8	N
64	GUN1	Date	8	N
65	NATOPS	Date	8	N
66	INST1	Date	8	N

67	INST2	Date	8		N
68	PQS	Date	8		N
69	RECCE	Date	8		N
70	LSO_REQUAL	Date	8		N
71	COURSE_RLS	Date	8		N
72	H2P_PQS	Date	8		N

J. Aircraft Database:

Field	Field Name	Type	Width	Dec	Index
* 1	BUNO	Numeric	6		Y
# 2	DET_NO	Numeric	2		Y
3	SIDE_NUMB	Numeric	4		N
4	AVAILABLE	Logical	1		N
5	MSN_STATUS	Character	6		N
6	FAM	Logical	1		N
7	SHIP	Logical	1		N
8	ASW	Logical	1		N
9	ASST	Logical	1		N
10	SAR	Logical	1		N
11	NIGHT	Logical	1		N
12	IMC	Logical	1		N
13	AMP_INFO	Character	30		N
14	LOCATION	Character	25		N
15	PMCF_REQ	Logical	1		N

APPENDIX B: FLIGHT SCHEDULING REGULATIONS

A. NATOPS Flight Manual:

1. Pilots must fly 30 hours in model in the previous twelve months of which 20 hours must be in the preceding six months to remain a current pilot qualified in model (PQM).

2. Airborne tactical officers (ATO) must fly 30 hours in model in the previous twelve months of which 20 hours must be in the preceding six months to remain a current ATO.

3. Aircrewmen must fly 50 hours as an ASW/ASST sensor operator with the preceding twelve months to remain current.

4. A qualified observer is an individual who has met all the minimum aeromedical and survival requirements for indoctrination flights set forth in OPNAVINST 3710.7 and has been thoroughly briefed.

5. To allow for qualification, a PQM may be substituted for a helicopter second pilot (H2P) on ASW/ASST and SAR/plane guard missions.

6. Minimum flight crew for an ASW/ASST mission is one helicopter aircraft commander (HAC), one ATO, and one ASW/ASST sensor operator (SO).

7. Minimum flight crew for a SAR mission is one HAC, one H2P or ATO, and two helicopter aircrewman (one of whom shall be SAR qualified).

8. Minimum flight crew for a utility mission is one HAC, one PQM, and one helicopter aircrewman.

9. Minimum flight crew for non-tactical/familiarization flights is two PQM's or one HAC and one qualified observer.

10. Minimum flight crew for flights from ships in the day or visual meteorological conditions (VMC) is two H2P's and one qualified aircrewman, or one HAC, one qualified observer, and one helicopter aircrewman.

11. Minimum flight crew for flights from ships at night or in instrument meteorological conditions (IMC) is one HAC, one PQM, and one aircrewman.

12. Minimum flight crew for instrument flight is one HAC, and one designated naval aviator (DNA), or two H2P's.

13. Minimum flight crew for functional check flights is one functional check pilot (FCP) and one qualified observer.

B. Squadron Pilot Training Syllabus:

1. Each pilot is required to have one hour of emergency procedure training per month.

2. OFT 1 and course rules exam are required prior to AC 1.

3. PQM's must have one day flight within 14 days of AC3.

4. Pilots must have completed at least one day doppler approach within the past seven days prior to being scheduled for AC 4.

5. One aircrewman is required during the simulated instrument portion of AC 5 and AC 6.

6. PQM's must have completed OFT 1, OFT 2, AC 1-7, and the H2P PQS prior to being scheduled for AC 8.

7. PQM's must successfully complete AC 8 prior to being scheduled for AC 9.

8. H2P's must be nominated for HAC and have completed the preliminary HAC open book test prior to being scheduled for AC 10.

9. H2P's must successfully complete AC 10 prior to being scheduled for AC 11.

10. Pilots must complete OFT 2 prior to being scheduled for SP 1 if NATOPS ship currency has expired.

C. Squadron Standard Operating Procedures:

1. Familiarization stage warm-up with a current HAC is required for any pilot who has not flown for a period of 30 calendar days or more.

2. Any pilot who has not flown at night within the last 30 days will be scheduled with a night current HAC on his next flight.

3. For night DLQ's, each pilot will have flown at night within the previous 15 days.

4. Aircrew need not report to the squadron until 10 hours prior to the scheduled completion of all flight and post flight duties.

5. Aircrew shall be allowed 10 hours in a non-duty status following their post flight responsibilities if those duties extend after 2200.

6. Pilots shall not be scheduled for a flight the day following Squadron Duty Officer (SDO) watch.

7. Aircrewmen shall not be scheduled for a flight if they have stood the 2400-0800 ASDO or security watch the same day.

8. Aircrewmen shall not be scheduled for a flight before 100 if they have stood the 1600-2400 watch the previous day.

9. Each pilot is required to complete three day hooded and three night coupled approaches every 30 days.

10. Only the HAC must be current for both pilots to conduct night coupled approaches.

D. Training and Readiness Manual:

All requirements with an asterisk (*) indicate that those qualifications if completed in a trainer are valid for only one half of the currency period of those done in the aircraft. In no case will the currency be less than six months.

1. ASW 1 through 6 which are valid for 12 months. (*)
2. ASW 7 and 8 which are valid for 12 months.
3. ASW 9 and 10 which are valid for 12 months. (*)
4. SURV 1 and 2 which are valid for six months. (*)
5. SURV 3 which is valid for six months.
6. EW 1 and 2 which are valid for six months. (*)
7. AAW 1 which is valid for six months. (*)

8. CCC 1 which is valid for six months. (*)

9. CCC 2 which is valid for 12 months.

10. CCC 3 which is valid for 12 months. (*)

11. SHIP 1 and 2 which are valid for 2 months.

12. SHIP 3 which is valid for 1 month.

13. SAR 1 and 2 which are valid for 1 month.

14. SAR 3 which is valid for 12 months.

15. FORM 1 which is valid for 6 months.

16. CGO 1 which is valid for 6 months.

17. NAV 1 which is valid for 6 months.

18. NAV 2 which is valid for 12 months.

19. HET 1 through 3 which are valid for 12 months.

20. GUN 1 which is valid for 1 month.

21. NATOPS which is valid for 12 months.

22. INST 1 which is valid for 1 month.

23. INST 2 which is valid for 12 months.

E. OPNAVINST 3710:

1. NATOPS evaluation may be renewed within 60 days preceaing expiration of a current evaluation and is valid for twelve months from the last day of the month in which the current evaluation expires. If there is no current evaluation, the evaluation is valid for 12 months from the last day of the month in which the evaluation is given.

2. Pilot's instrument rating must be renewed prior to the end of the birth month and no sooner than 60 days prior to the first day of the birth month.

3. Aircrew annual flight physical must be renewed plus or minus 30 days of the aircrewman's birthday.

4. Aviation physiology training is valid for 4 years.

5. Water survival training is valid for 4 years.

6. No flight duties for twelve hours after undergoing Naval Aviation water survival training program.

F. Heuristics:

1. Pilots preparing to deploy have precedence for all night and shipboard flights if they are not qualified.

2. NATOPS and instrument proficiency checks have a high priority and they ideally shall be completed no later than 15 days prior to their expiration.

3. All deploying detachment pilots must have flown at night within the previous 15 days.

4. Emergency egress must be completed every year.

5. Aircraft that require functional check flights should not be scheduled for any other missions due to the uncertainty of when the check flight will be complete.

6. Detachment aircrew should be scheduled to fly together whenever possible for crew coordination training.

7. Only designated NATOPS instructors can administer NATOPS proficiency checks.

8. Only designated special instrument pilots can administer instrument proficiency checks.

9. Only designated and current landing signal officers (LSO) can be scheduled as LSO.

10. Pilots that possess higher ratings such as NATOPS instructor should not be scheduled for other training missions if that higher rating is required for a check flight.

11. Scheduled takeoff times should always account for the transit time required to arrive on station to complete the assigned mission.

12. Whenever an aircraft is to remain turning during a crew change, the flight scheduler should plan on 30 minutes prior to the next crew's takeoff.

13. Whenever an aircraft is to be secured prior to the next flight, the flight scheduler should plan on a minimum of one hour prior to the next crew's takeoff to account for all necessary maintenance and inspections.

14. The normal priority of missions in descending order are those that are directed by higher authority, those required to meet detachment's underway periods, NATOPS and instrument proficiency check flights, HAC and H2P check flights, and general squadron training flights.

APPENDIX C: DATABASE PROTOTYPE PROGRAMS

```
**************************JPROCLIB.PRG
*---Custom dBASE IV procedures and functions for THESIS

*---Procedure to center any character string using any right margin
PROCEDURE Center
   PARAMETERS Title, RMargin
       Padding = SPACE((RMargin/2)-LEN(TRIM(Title))/2)
       ? Padding+TRIM(Title)
RETURN

**************************Automatically assign aircrew numbers
PROCEDURE AUTOACNO
   *......Open Aircrew Database
   USE Aircrew ORDER Aircrew_No
   GOTO BOTTOM
   *......1001 is smallest possible aircrew number
   Largest = MAX(1000,Aircrew_No)
   NextAC = Largest + 1
   *......Fill in aircrew numbers
   USE Aircrew  && Deactivate index before using replace
   SCAN FOR Aircrew_No < 1000
         REPLACE Aircrew_No WITH NextAC
         NextAC = NextAC + 1
   ENDSCAN
   CLOSE DATABASES
RETURN

**************************Automatically assign mission numbers
PROCEDURE AUTOMSNO
   *......Open Mission Database
   USE Mission ORDER Mission_No
   GOTO BOTTOM
   *......10 is smallest possible mission number
   Largest = MAX(10,Mission_NO)
   NextMsn = Largest + 1
   *..... Fill in mission numbers
   USE Mission  && Deactivate index before using replace
   SCAN FOR Mission_No < 10
         REPLACE Mission_No WITH NextMsn
         NextMsn = NextMsn + 1
   ENDSCAN
   CLOSE DATABASES
RETURN
```

```
***************************Rebuild  index  files  for  all  Thesis
databases
PROCEDURE Thsrendx
   SET TALK ON     && Show progress
   USE Event
   REINDEX
   USE Trainers
   REINDEX
   USE Aircraft
   REINDEX
   USE Aircrew
   REINDEX
   USE Quals
   REINDEX
   USE Sched
   REINDEX
   USE Mission
   REINDEX
   USE Training
   REINDEX
   SET TALK OFF    && Suppress program messages
RETURN

**************************Opens database files and sets relations
**************************for pilot and aircrew snivels
PROCEDURE Snivel
   SELECT A
   USE Sched
   SELECT B
   USE Aircrew ORDER Aircrew_No
   SELECT Sched
   SET RELATION TO Aircrew_No INTO Aircrew
   GO TOP
RETURN

**************************Validate Aircrew_Number for pilot snivel
FUNCTION ISAC
PARAMETER Myacno
   DO CASE
      *If user doesn't enter number, do nothing
      CASE Myacno = 0
         OK = .T.
      *If aircrew number was entered
      CASE SEEK(Myacno,"Aircrew")
         @ 9,30 SAY Aircrew->Last_Name
         @ 10,30 SAY Aircrew->First_Name
         @ 10,61 SAY Aircrew->MI
         OK = .T.
      OTHERWISE
```

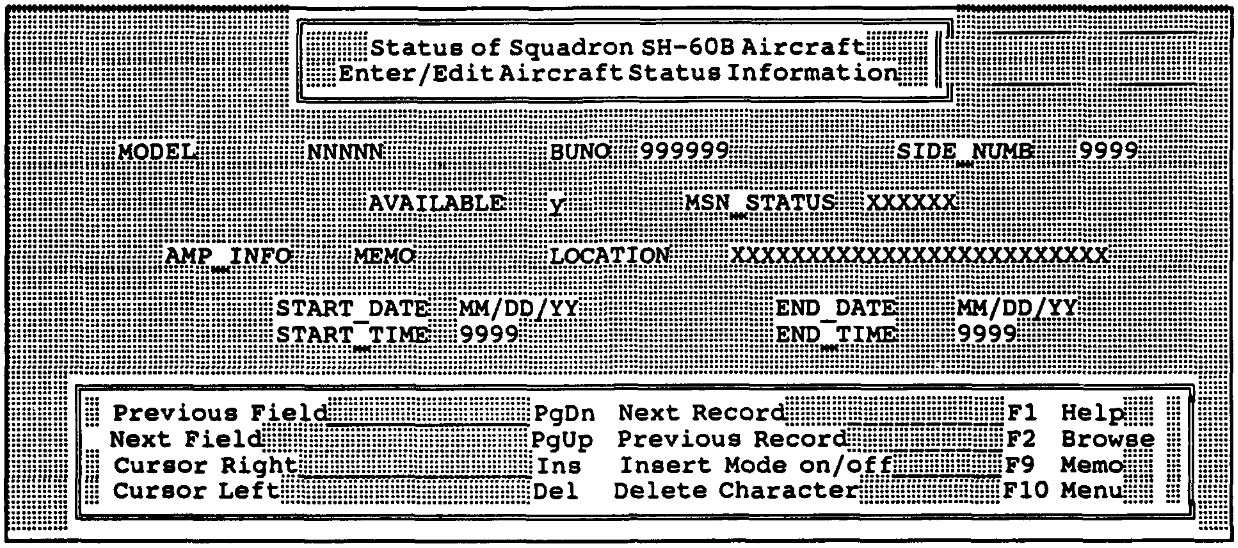
```
         @ 6,43 SAY "No such Aircrew!"
         @ 6,43 SAY SPACE(25)
         OK = .F.
   ENDCASE
RETURN (OK)

******************************Validates entered time
FUNCTION TIMECK
PARAMETER Mytime
    DO CASE
      CASE Mytime < 100
         OK = .F.
      CASE Mytime > 2400
         OK = .F.
      Case MOD(Mytime,100) <> 0
         OK = .F.
      OTHERWISE
         OK = .T.
   ENDCASE
RETURN (OK)

**************************Determine aircrew availability based on
date and time
PROCEDURE AVAILABILITY
PARAMETERS Mdate, Mtime
   SELECT A
   USE Sched
   SELECT B
   USE Aircrew ORDER Aircrew_No
   SELECT Sched
   SET RELATION TO Aircrew_No INTO Aircrew
      SCAN FOR Aircrew_NO > 1000
         DO CASE
            CASE Mdate >= Start_Date .AND. Mdate < End_Date
               REPLACE Aircrew->Available WITH .N.
            CASE Mdate = End_Date .AND. Mtime < End_Time
               REPLACE Aircrew->Available WITH .N.
         ENDCASE
      ENDSCAN
CLOSE DATABASES
RETURN
```

APPENDIX D: EXAMPLES OF DATABASE FORMS

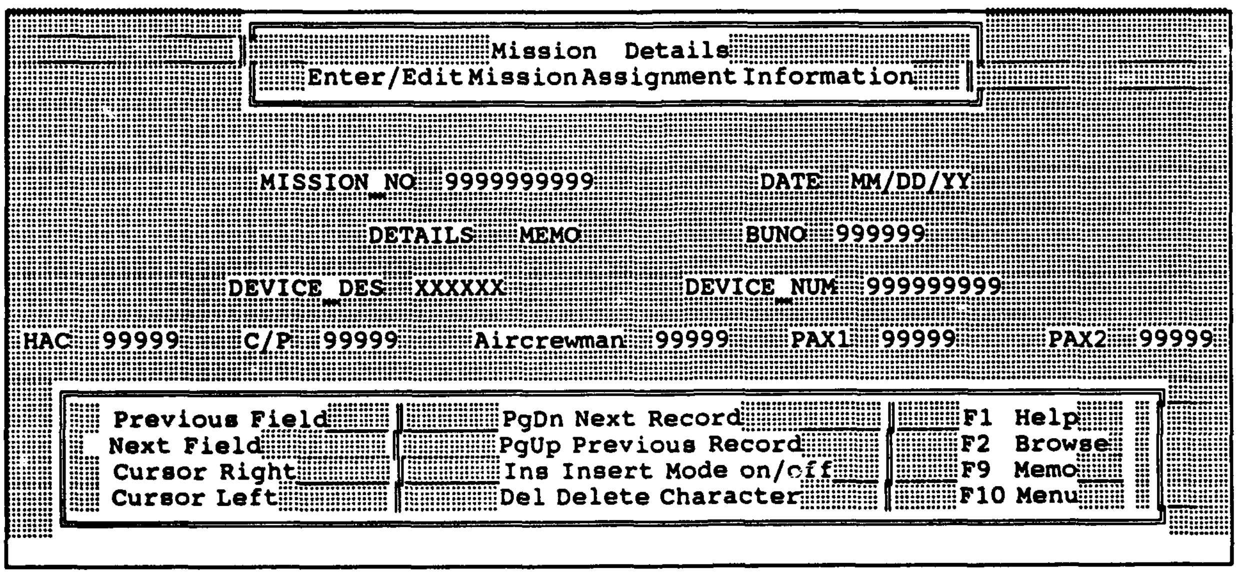
1. Aircraft Availability Form:


Status of Squadron SH-60B Aircraft
Enter/Edit Aircraft Status Information

MODEL NNNNN BUNO 999999 SIDE_NUMB 9999
AVAILABLE y MSN_STATUS XXXXXX
AMP_INFO MEMO LOCATION XXXXXXXXXXXXXXXXXXXXXXXX
START_DATE MM/DD/YY END_DATE MM/DD/YY
START_TIME 9999 END_TIME 9999

Previous Field PgDn Next Record F1 Help
Next Field PgUp Previous Record F2 Browse
Cursor Right Ins Insert Mode on/off F9 Memo
Cursor Left Del Delete Character F10 Menu

2. Mission Information Form:


Mission Details
Enter/Edit Mission Assignment Information

MISSION_NO 9999999999 DATE MM/DD/YY
DETAILS MEMO BUNO 999999
DEVICE_DES XXXXXX DEVICE_NUM 999999999
HAC 99999 C/P 99999 Aircrewman 99999 PAX1 99999 PAX2 99999

Previous Field PgDn Next Record F1 Help
Next Field PgUp Previous Record F2 Browse
Cursor Right Ins Insert Mode on/off F9 Memo
Cursor Left Del Delete Character F10 Menu

3. Pilot Data Form

```
Pilot Personnel Data
Enter/Edit Data for Squadron Pilots

AIRCREW_NO 99999    LAST_NAME   AAAAAAAAAAAAAAAAAAAAAAAAA

                    FIRST_NAME  AAAAAAAAAAAAAAAAAAAAAAAAA        MI !

                    SSN         999-99-9999    BIRTHDAY   MM/DD/YY

                    DESIG       AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Previous Field      PgDn  Next Record              F1  Help
Next Field          PgUp  Previous Record          F2  Browse
Cursor Right        Ins   Insert Mode On/Off       F10 Menu
Cursor Left         Del   DeleteCharacter
```

4. Pilot Qualification Record Form

```
Squadron Pilot's Sh-60B/Trainer Qualifications
Enter/Edit Date that qualification was earned

                         AIRCREW_NO 99999

           PQM          MM/DD/YY     H2P          MM/DD/YY

           ATO          MM/DD/YY     LSO          MM/DD/YY

           HAC          MM/DD/YY     FCP          MM/DD/YY

           NATOPSINST   MM/DD/YY     INSTR_INST   MM/DD/YY

OFT_INST   MM/DD/YY     WTT_INST     MM/DD/YY          WST_INST   MM/DD/YY

Previous Field      PgDn Next Record          F1  Help
Next Field          PgUp Previous Record      F2  Browse
```

5. Aircrew Snivel Form

```
                    Aircrew  Snivel
         Enter/Edit Pilot Availability Information

                    AIRCREW_NO  99999

        LAST_NAME   XXXXXXXXXXXXXXXXXXXXXXXXX
        First_Name  XXXXXXXXXXXXXXXXXXXXXXXXX   MI  X

Starting Date of Snivel  MM/DD/YY     Ending Date of Snivel  MM/DD/YY
Starting Time of Sinvel  9999         Ending Time of Snivel  9999

 Previous Field        Pg Dn  Next Record            F1  Help
 Next Field            Pg Up  Previous Record        F2  Browse
 Cursor Right          Ins    Insert Mode on/off     F10 Menu
```

APPENDIX E: EXPERT SYSTEM PROTOTYPE PROGRAM CODE

```
!---------------------Statements Block

RUNTIME;

ENDOFF;

AUTOQUERY;

BKCOLOR = 3;

ASK Schedule_Date:

"Enter the date you wish to schedule for in the following format:
19YearMonthDay  ie. 19910901";

ASK Database_Status:
"Are you sure that you have all the necessary information to
commence flight scheduling? Has that information been updated and
is it current for this date: {Schedule_Date}?";

CHOICES Database_Status, Continue, Msnagain, Acftagain: YES, NO;

ASK Continue: "Do you still want to continue this consultation with
the  LAMPS^GMK III Expert Flight Scheduling Program?";

ASK Mission: "These are the missions which have dates on
{Schedule_Date}. Which Mission do you want to schedule now?";

ASK Msnagain: "Would you like to schedule another mission?";

ASK Platform: "Do you want an Aircraft, or a Trainer for this
mission?";

CHOICES Platform: Aircraft, Trainer;

ASK SchedAircraft: "These are the BUNO of the squadron aircraft
available on {Schedule_Date} and during the scheduled mission time
of {Mstrt} - {Mend}.
Select the one you want to schedule for this mission.";

ASK Position: "What crew position do you want to schedule this
pilot for?";

CHOICES Position: HAC, CP, SO, PAX, Instructor;
```

```
ASK Acftagain: "Do you want to select another aircraft?";


!---------------------Actions Block

ACTIONS

COLOR = 0

WOPEN 2,2,5,12,70,4

ACTIVE 2

DISPLAY "    Welcome to the LAMPS MK III Expert Flight Scheduling
Program!


This is Version 1.0 written in August 1991.  It will assist you to
make optimal flight scheduling decisions based on squadron database
information, and your answers to the program's questions.
Please refer any proposed improvements to LCDR John O'Connor.


Press any key to begin the program.........~"

WCLOSE 1

FIND Schedule_Date

CLS

FIND Database_Status

WHILETRUE Database_Status = No THEN

   WOPEN 3,2,5,9,68,4

   ACTIVE 3

   DISPLAY

"Flight Scheduling isn't recommended until the 8 squadron
databases:(1)Aircrew.DBF, (2) Aircraft.DBF, (3) Quals.DBF, (4)
Training.DBF,(5) Sched.DBF, (6) Trainers.DBF, (7) Det.DBF, and (8)
Daytime.DBF have been updated.  You should be confident that the
information they are storing is accurate for {Schedule_Date}, or
your flight schedule will probably be in error!"
```

```
   GETCH Temporary

   WCLOSE 3

   RESET Database_Status

   FIND Continue

   RESET Temporary

END

WHILETRUE Continue = Yes OR Database_Status = Yes THEN

   RESET Continue

   RESET Database_Status

   Msnagain = Yes

END

WHILETRUE Msnagain = Yes THEN

   RESET Msnagain

   CLS

   MENU Mission, Schedule_Date = Date, E:\DBASE2\THESIS\Mission,
Mission_No

   FIND Mission

   MRESET Mission

   GET  MISSION  =  Mission_No  AND  Schedule_Date  =  Date,
E:\DBASE2\THESIS\Mission, ALL

   CLS

   DISPLAY "This is a summary of the mission you selected:"

   DISPLAY" "

   DISPLAY

"MSN #  Datc         Start     End       Mission"

   COLOR = 14
```

```
   DISPLAY
"{Mission_No}       {Date}      {Mstrt_Time}          {Mend_Time}
{Msn_Type}"
   DISPLAY" "
   DISPLAY
   "Previously Scheduled: {Scheduled}
Location:  {Location}
Remarks:   {Added_Info}"
   COLOR = 0
   DISPLAY " "
   DISPLAY " "
   DISPLAY " "
   FIND Msnagain
Thismission = Found
END
FIND Platform

WHILETRUE Platform = Aircraft THEN
   FIND Acftcheck
END

WHILETRUE Platform = Trainer THEN
   FIND Trainercheck
END
```

```
WHILETRUE Platform = Aircraft AND Acftcheck = Right AND Thismission
= Found THEN

    RESET Platform

    RESET Acftcheck

    CLS

    MENU SchedAircraft, Schedule_Date = (Sstrt_Date) OR
Schedule_Date = (Send_Date) AND Mstrt_Time >= (Sstrt_Time) AND
Mend_Time <= (Send_Time), E:\DBASE2\THESIS\Sched, Buno

    FIND SchedAircraft

    MRESET SchedAircraft

    GET SchedAircraft = BUNO, E:\DBASE2\THESIS\Aircraft, ALL

    GET SchedAircraft = BUNO, E:\DBASE2\THESIS\Sched, ALL

    DISPLAY "This is the information on the aircraft you selected:"

    DISPLAY" "

    COLOR = 14

    DISPLAY

"BUNO           = {BUNO}

Side Number    = {Side_Numb}

Availability   = {Available}

Mission Status = {Msn_Status}

FAM Capable    = {FAM}            Starting Date = {Sstrt_Date}

Ship Capable   = {Ship}            Starting Time = {Sstrt_Time}

ASW Capable    = {ASW}            Ending Time   = {Send_Time}

ASST Capable   = {ASST}            Ending Date   = {Send_Date}

SAR Capable    = {SAR}

Night Capable  = {Night}

IMC Capable    = {IMC}
```

```
Remarks         = {Amp_Info}
Location        = {Location}
PMCF Required   = {PMCF_Req}"
    COLOR = 0
    DISPLAY "Press any key to continue......~"
    CLS
    FIND Acftagain
    WHILETRUE Acftagain = Yes THEN
       RESET Acftagain
       FIND Platform
       FIND Acftcheck
       RESET SchedAircraft
       RESET Sstrt_Date
       RESET Sstrt_Time
       RESET Send_Date
       RESET Send_Time
       RESET Aircrew_No
       RESET Buno
       RESET Det_No
       RESET Side_Numb
       RESET Available
       RESET Msn_Status
       RESET Fam
       RESET Ship
       RESET ASW
       RESET ASST
```

```
        RESET SAR
        RESET NIGHT
        RESET IMC
        RESET AMP_Info
        RESET Location
        RESET PMCF_Req
        CLOSE E:\DBASE2\THESIS\Aircraft
        CLOSE E:\DBASE2\THESIS\Sched
END
 CLOSE E:\DBASE2\THESIS\Mission

 RESET Mission
DISPLAY "End of program.  Select Go to run again."
DISPLAY "Select Quit (twice) to return to DOS";

!---------------------------------RULES BLOCK

RULE 1
IF          Platform = Aircraft
            AND Msn_Type = Trainer
            OR Msn_type = ASW_Rodeo
THEN
            WOPEN 3,15,5,3,68,4
            ACTIVE 3
            DISPLAY "      The mission requires a Trainer, not an
Aircraft!!"
```

```
            GETCH Temporary
            RESET Temporary
            WCLOSE 3
            Acftcheck = Wrong
ELSE
            Acftcheck = Right;


RULE 2
IF          Platform = Trainer
            AND Msn_Type = Logistics
            OR Msn_Type = HET
            OR Msn_Type = Day_Bits
            OR Msn_Type = Night_Bits
            OR Msn_Type = SAR
THEN
            WOPEN 4,15,5,3,68,4
            ACTIVE 4
            DISPLAY "      The mission requires an Aircraft, not a
Trainer!!"
            GETCH Temporary
            RESET Temporary
            WCLOSE 4
            Trainercheck = Wrong
ELSE
            Trainercheck = Right;
```

LIST OF REFERENCES

Awad, E. M., *Management Information Systems*, The Benjamin Cummings Publishing Company, Inc., 1988.

Chief of Naval Operations, Naval Air Systems Command, *NATOPS Flight Manual Navy Model SH-60B Aircraft A1-H60BB-NFM-000*, 01 September 1987.

Department of the Navy, Anti-Submarine Warfare Wing Pacific Fleet, *Training and Readiness Manual COMASWWINGPACINST 3500.60A*, 08 April 1991.

Department of the Navy, Helicopter Anti-Submarine Squadron Light Forty Three, *Pilot Training Syllabus HSL43INST 3501.1A*, 21 August 1989.

Department of the Navy, Helicopter Anti-Submarine Squadron Light Forty Three, *Aircraft Standard Operating Procedures HSL43INST 3710.1F*, 04 March 1991.

Department of the Navy, Office of the Chief of Naval Operations, *NATOPS General Flight and Operating Instructions OPNAVINST 3710.7N*, 10 April 1990.

Dolan, K., Kroenke, D. M., *Database Processing Fundamentals Design Implementation*, Macmillan Publishing Company, 1988.

Feigenbaum, E., McCorduck, P., and Nii, H. P., *The Rise of the Expert Company*, First Vintage Books, October 1989.

Interview between T. Jara, LCDR, USN, Helicopter Anti-Submarine Squadron Light Forty Three, San Diego, California, and the author, 03 July 1991.

Interview between F. Bosio, CDR, USN, Training Squadron 26, Beeville, Texas, and the author, 24-25 June 1991.

Jackson, P., *Introduction to Expert Systems*, 2d ed., Addison-Wesley Publishing Company, 1990.

Page-Jones, M., *The Practical Guide to Structured Systems Design*, 2d ed., Yourdon Press, 1988.

Rolston, D. W., *Principles of Artificial Intelligence and Expert Systems Development*, R. R. McGraw-Hill Book Company, 1988.

Turban, E., *Decision Support and Expert Systems*, 2d ed., Macmillan Publishing Company, 1990.

BIBLIOGRAPHY

Awad, E. M., *Management Information Systems*, The Benjamin Cummings Publishing Company, Inc., 1988.

Bauman, J., and others, *Using dBase IV*, Que Corporation, 1990.

Bloom, S., *Using DrawPerfect*, Que Corporation, 1991.

Campbell, M., *Harvard Graphics Made Easy*, Osborne McGraw-Hill, 1990.

Carlson, E. D., and Sprague, R. J., *Building Effective Decision Support Systems*, Prentice-Hall, 1982.

Chief of Naval Operations, Naval Air Systems Command, *NATOPS Flight Manual Navy Model SH-60B Aircraft A1-H60BB-NFM-000*, 01 September 1987.

Department of the Navy, Anti-Submarine Warfare Wing Pacific Fleet, *Training and Readiness Manual COMASWWINGPACINST 3500.60A*, 08 April 1991.

Department of the Navy, Helicopter Anti-Submarine Squadron Light Forty Three, *Pilot Training Syllabus HSL43INST 3501.1A*, 21 August 1989.

Department of the Navy, Helicopter Anti-Submarine Squadron Light Forty Three, *Aircraft Standard Operating Procedures HSL43INST 3710.1F*, 04 March 1991.

Department of the Navy, Office of the Chief of Naval Operations, *NATOPS General Flight and Operating Instructions OPNAVINST 3710.7N*, 10 April 1990.

Dolan, K., and Kroenke, D. M., *Database Processing Fundamentals Design Implementation*, Macmillan Publishing Company, 1988.

EasyCase Plus Version 2.0 Release Notes, Evergreen Case Tools, 1990.

Feigenbaum, E., McCorduck, P., and Nii, H. P., *The Rise of the Expert Company*, First Vintage Books, October 1989.

Hamblen, D., "By Aviators for Aviators," *CHIPS*, v. XI Issue 1, pp. 4-6, January 1991.

Hicks, R., and Lee, R., *VP-Expert for Business Applications*, Holden-Day, Inc., 1988.

Interview between T. Jara, LCDR, USN, Helicopter Anti-Submarine Squadron Light Forty Three, San Diego, California, and the author, 03 July 1991.

Interview between F. Bosio, CDR, USN, Training Squadron 26, Beeville, Texas, and the author, 24-25 June 1991.

Jackson, P., *Introduction to Expert Systems*, 2d ed., Addison-Wesley Publishing Company, 1990.

Page-Jones, M., *The Practical Guide to Structured Systems Design*, 2d ed., Yourdon Press, 1988.

Rolston, D. W., *Principles of Artificial Intelligence and Expert Systems Development*, R. R. McGraw-Hill Book Company, 1988.

Senn, J. A., *Analysis and Design of Information Systems*, 2d ed., McGraw-Hill Publishing Company, 1989.

Sharp, M., and Stark, R., *dBase IV 1.1 The Master Reference*, Windcrest Books, 1990.

Simpson, A., *Mastering WordPerfect 5.1*, Sybex Inc., 1990.

Simpson, A., *Understanding dBase IV*, Sybex Inc., 1989.

Turban, E., *Decision Support and Expert Systems*, 2d ed., Macmillan Publishing Company, 1990.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	LCDR John B. O'Connor Helicopter Anti-Submarine Squadron Light Forty Three Naval Air Station North Island San Diego, California 92135-5207	2
4.	Professor Martin J. McCaffrey, Code AS/Mf Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
5.	Computer Technology Code 37 Naval Postgraduate School Monterey, California 93943-5000	1