AIR COMMAND
AND
STAFF COLLEGE

STUDENT REPORT

APPLICABILITY OF
ARTIFICIAL INTELLIGENCE TO
THE GLOBAL DECISION SUPPORT SYSTEM

MAJOR RICHARD J. BLANCHET 87-0245

"insights into tomorrow"

DISCLAIMER

REPORT NUMBER 87-0245

TITLE APPLICABILITY OF ARTIFICIAL INTELLIGENCE TO THE GLOBAL DECISION SUPPORT SYSTEM

AUTHOR(S) MAJOR RICHARD J. BLANCHET

FACULTY ADVISOR CAPT RONALD FORD, ACSC/XP

SPONSOR BRIG GEN ROBERT V. WOODS, HQ MAC ASST DCS/OPERATIONS

Submitted to the faculty in partial fulfillment of requirements for graduation.

AIR COMMAND AND STAFF COLLEGE
AIR UNIVERSITY
MAXWELL AFB, AL 36112

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A180453

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT STATEMENT "A" Approved for public release; Distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 87-0245	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION ACSC/EDCC; 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Maxwell AFB, AL 36112-5542	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
APPLICABILITY OF ARTIFICIAL INTELLIGENCE TO THE GLOBAL DECISION SUPPORT SYSTEM

12. PERSONAL AUTHOR(S)
Blanchet, Richard J., Major, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
	FROM ____ TO ____	1987 April	38

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

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

The Global Decision Support System (GDSS) is a prototype command and control center for the Military Airlift Command (MAC). This computer network links the top three levels of MAC's command structure, and as changes are made, all other nodes are updated automatically. This study evaluates the suitability of incorporating artificial intelligence (AI) with GDSS. The basics of AI "expert systems" and Natural Language Processors is examined. The study concludes that while technically feasible, integration would be operationally impractical until certain technical and organizational milestones are achieved.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL ACSC/EDCC Maxwell AFB, AL 36112-5542	22b. TELEPHONE (Include Area Code) (205) 293-2483	22c. OFFICE SYMBOL

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

INTRODUCTION

"Artificial Intelligence is the most pervasive technology underlying the vision of the future" as seen by Brig Gen Charles F. Stebbins, Deputy Chief of Staff for Science and Technology, Air Force Systems Command, after completing Project Forecast II (10:70). Is artificial intelligence (AI) a way to capture and improve on the corporate knowledge frequently lost through transfers and retirements? In a strongly worded statement, Dr. Bernard Kulp, Air Force System Command's Chief Scientist said, "The applications of AI are likely to be as extensive as the applications of the digital computer" (10:70). If this is true, big changes are around the corner.

The purpose of this paper is to investigate whether artificial intelligence should be integrated with the Global Decision Support System (GDSS). To accomplish this goal, Chapter 1 describes what GDSS is, how it works, and why GDSS was developed in the first place. Simply stated, GDSS is an attempt to computerize the command and control functions of the Military Airlift Command's top three echelons. This computer network passes data automatically to all of the other "nodes," or participants of the system. The reason GDSS was developed is the critical need to improve the command and control of irreplaceable airlift assets and their crews. Once the basic concept of the operation and scope of GDSS is covered, the subject shifts to artificial intelligence.

In Chapter 2, artificial intelligence is defined along with a brief description of its background and characteristics. This second chapter focuses on two specific areas of AI, namely, expert systems and Natural Language Processors (NLP), which potentially apply to GDSS. An expert system is a type of computer program that is able to solve problems normally associated with "intelligent" behavior. However, an exact definition of an expert system simply does not exist. A Natural Language Processor allows the computer operator to interact with a computer by using normal english commands, just as if they were talking with another human. At the end of Chapter 2 is a discussion of some of the problems associated with artificial intelligence.

CONTINUED

Once the basis of GDSS and AI have been established, Chapter 3 ties them together. First, if artificial intelligence should be integrated with GDSS, there needs to be strong rationale. Simply worded, the reason is to make GDSS operate more effectively in command and control of airlift assets. This motive is developed in the section dealing with the basis of change and time value of data. Applications of artificial intelligence to GDSS start with the need for an adaptable interface with GDSS, the conversion of GDSS software to the ADA computer language, and finally AI's use against "backdoors" and "worms." Following this are specific recommendations for integration of AI to the Global Decision Support System. These recommendations focus on two areas: technical and organizational problems. Lastly, and most importantly, Chapter 3 specifies the prerequisites for the successful integration of the two systems.

ACKNOWLEDGEMENTS

Many people have contributed their time and creativeness in the development of this study. Foremost thanks goes to Lt Col Richard Poff and Capt Douglas Story (MAC HQ/DOA) whose drive and resourcefulness keep the Global Decision Support System running, while also supporting a "stranger" from Air University. Mr Murray Daniels of the MITRE Corp has also dispensed a great deal of insight to the programming complexities of AI. Some select people from Travis AFB nurtured a "common sense approach" to automated command and control, years ahead of its time. The Wing Commander (Brig Gen Robert Woods), and the Deputy for Operations (Col Jay Baughman) stuck their necks (and wallets) out to support the crazy ideas of a couple of captains. Capt Glenn Bailey rates a special thank-you as a co-developer of the Travis Command and Control Display System, and as a "sounding board" for numerous other computer concepts and hairbrained ideas. Finally, this study is also an indirect product of my wife, Susan, whose concern, patience, and good humor are my biggest asset.

ABOUT THE AUTHOR

Major Richard J. Blanchet graduated from the USAF Academy in 1974 with a Bachelor of Science degree in Behavioral Science. After completing USAF Pilot training at Columbus AFB, Mississippi, he remained there as a T-37 Instructor/Check Pilot until 1978. Following this assignment, Major Blanchet flew the C-9A Nightingale (an Aeromedical Evacuation aircraft) based at Scott AFB as an Aircraft Commander/Evaluation Pilot. In 1983, he transferred to the C-5A Galaxy at Travis AFB, California. Here, as an air-refueling qualified aircraft commander, he was awarded the 5000 hour flying safety award presented by the Military Airlift Command. Additionally, Major Blanchet has completed a Masters of Business Administration (MBA) degree in Management with Golden Gate University.

While at Travis AFB as a Command Post Duty Officer, in collaboration with Captain Thomas G. Bailey, he designed the hardware and wrote the necessary software for a multiuser, multitasking computer system used in the command and control of the Military Airlift Command's largest wing--the 60th MAW. The system was tested and proven in several ORI's and large scale DOD exercises. Major Blanchet is a published author of several medical journal articles on aerospace medicine and physiology.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

EXECUTIVE SUMMARY

REPORT NUMBER 87-0245

AUTHOR(S) MAJOR RICHARD J. BLANCHET

TITLE APPLICABILITY OF ARTIFICIAL INTELLIGENCE TO THE GLOBAL DECISION SUPPORT SYSTEM

I. <u>PROBLEM:</u> Should an artificial intelligence (AI) expert system be integrated into the Global Decision Support System (GDSS)?

II. <u>OBJECTIVES:</u> First objective: look at the requirements and goals of the Global Decision Support System. It answers the question: What is GDSS? Second objective: explain artificial intelligence and expert systems. Included in this discussion is information on Natural Language Interfaces, a particular type of expert system. Finally, the specific areas artificial intelligence could strengthen GDSS (with the appropriate milestones to be met prior to its integration) are presented.

III. <u>DISCUSSION OF ANALYSIS:</u> GDSS is a sophisticated computer system under accelerated development, designed to automate the Military Airlift Command's top three levels of command and control. Specifically, these are MAC HQ, 21st Air Force, 22nd Air Force, 834th Air Lift Division, and the 322nd Air Lift

CONTINUED

Division. Each location has a computer local area network, and the software necessary to perform their command and control tasks. All locations are interconnected via a Wide Area Network (WAN) so that as inputs are made, the other locations are automatically updated.

Artificial intelligence has been identified in Project Forecast II as a technical development that is predicted to revolutionize the way the Air Force does it's job. It is basically software programs able to perform tasks normally associated with intelligence and reasoning. This process is achieved by the use of a large knowledge base interacting with "rules of thumb" as the data base is manipulated. Several AI systems have demonstrated this intelligent behavior, but is generally limited to specific domains of knowledge, such as medicine. The U.S. Navy is currently using several AI programs. Some AI systems use a Natural Language Processor which is a program that allows the user to interact with the computer in a non-structured method. For example, questions can be posed to a computer in normal english syntax.

IV. <u>CONCLUSIONS:</u> AI can be integrated into GDSS to improve both its operation and its maintainability. However, this must be held in check until several specific technical and organizational milestones have been met. Specific technical prerequisites are improvements in: knowledge base development, AI programmer knowledge/availability, speed/parallel microprocessor, and Natural Language Processors. Organizational milestones include education of "blue suiters" in AI to monitor design and development. Additionally, there is a critical need to implement an adaptable interface with GDSS capable of being deployed worldwide. Finally, AI would help to resolve conflicts of data and improve user acceptance.

VI. <u>RECOMMENDATIONS:</u> AI programs should <u>NOT</u> be integrated with GDSS until the specific milestones examined are achieved. This is based on the current level of the technology and budgetary realities. The objectives of GDSS to improve the command and control of airlift operations can be greatly enhanced through AI. However, there is presently a large gap between technically feasible uses of AI and operationally attainable ones.

Chapter One

WHAT IS GDSS?

The aim of this chapter is to explain what the Global Decision Support System (GDSS) is, why it is being developed, and briefly summarize its current status. It is not intended to illustrate the technical aspects and design of the system but develop the basis for the concept--its reason for being. To accomplish this task, this chapter will cover the program's objectives, a description of the hardware, and the software involved. An essential part of the chapter answers the question: Why is GDSS so critical? To begin, let's take a look at some program objectives to explain what the Global Decision Support System really intends to do.

PROGRAM OBJECTIVES

According to the GDSS Task Implementation Plan, the Jet Propulsion Laboratory (JPL) will develop a command and control system that will:

1. Enhance Military Airlift Command (MAC) [sic] operational capability.
2. Act as an evolutionary step in the overall command and control upgrade program.
3. Provide the DOD with a prototype command center system (18:1-1).

To satisfy these objectives, JPL has developed a computer system currently being installed, but not operational, to demonstrate state-of-the-art command and control of the top three MAC echelons. These three echelons are; MAC Headquarters, Scott AFB; 21 AF, McGuire AFB; 22 AF, Travis AFB; 834 ALD, Hickham AFB; and 322 ALD, Ramstein AB (16:1-1). Scott AFB, Travis AFB, and McGuire AFB have Digital Equipment Corporation (DEC) VAX 8600 computers as the main computers (See Figure 1). The two overseas Air Lift Divisions (ALD) have DEC MicroVax II computers connected to form a local area network. In order to connect each of these geographic locations (nodes), a wide area network (WAN) is used. A wide area network is simply the means used to connect several local area networks together. Speed is usually reduced in transmission rates as compared to a local area net. The WAN used by the Jet Propulsion Laboratory to connect the local networks is the Ethernet Datagram Service Translan software (18:10-1). As you can see in Figure 1, GDSS has five geographically separated locations, each with its own local network, connected via a WAN, and using the same data base.

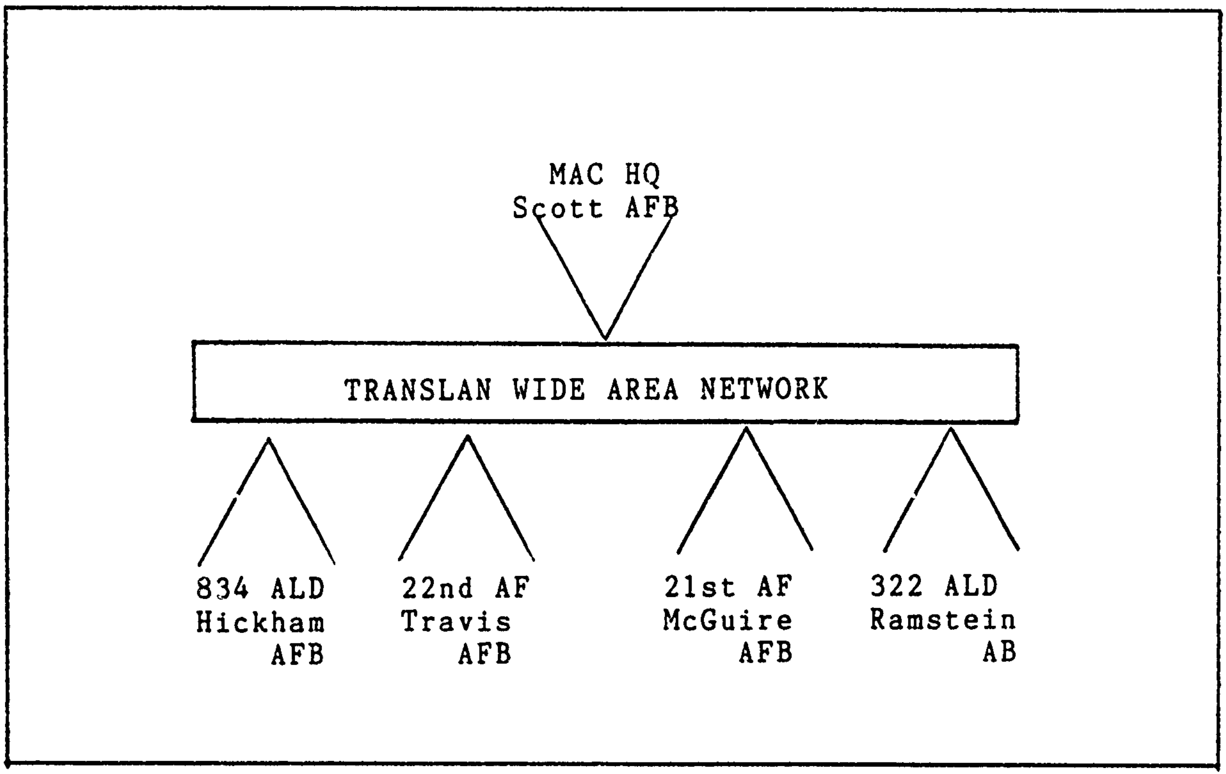

Figure 1
Diagram of GDSS Geography

When one location makes a change to the data base, all the other locations are updated. To illustrate, at each location, a terminal operator interacts with a computer program developed for a specific organization, such as the command post. The information would then be passed to both the local data base management software and network interface. The information is passed along to the additional terminals locally and the other echelons via the wide area network. Each organization has to have its own software programs to support its specific needs, while also interfacing with the data being supplied from the other organizations. For example, the Command Post program has 32,000 lines of code written in FORTRAN, as of October 17, 1986 (17:--). Current Operations, Plans, Logistics, Transportation, and the Crisis Action Team's programs are also under development in computer languages, such as COBAL and ADA. The size of their program code varies with functional area requirements. This is a major software undertaking which calls for extensive custom programing. The software will be discussed further in Chapter 3, where GDSS and artificial intelligence are linked. As the GDSS program becomes larger, there will be "bugs" in the software programs. To counter this, JPL has planned four major upgrades of the system software during the three year project. To complete this aggressive tasking, JPL adopted this approach:

> The implementation must make use of off-the-shelf hardware and software to the extent possible, and accommodate a wide range of user involvement in defining and evaluating the effectiveness of the system. The system will not preclude off-the-shelf software upgrades. Such development is well suited to an evolutionary development. An evolutionary development is an acquisition strategy in which a basic or critical capability is acquired initially and fielded quickly, based on a short need statement that includes a representative description of the overall capability needed and the architectural framework within which evolution will occur (16:1-11 - 1-12).

This evolutionary approach is the basis for the changes to be described in the third chapter and shows the original design intent for improvements to the Global Decision Support System. To see where GDSS is, in its development process, refer to Figure 2.

As a recap, there are five local networks connected and sharing common data. For example, when Travis AFB enters the departure of an aircraft, the data base at Scott AFB and McGuire AFB are also automatically updated. This insures an accurate duplication of information since all other nodes in the system have also been changed. The preceding discussion has addressed the "what" of GDSS; now for the why.

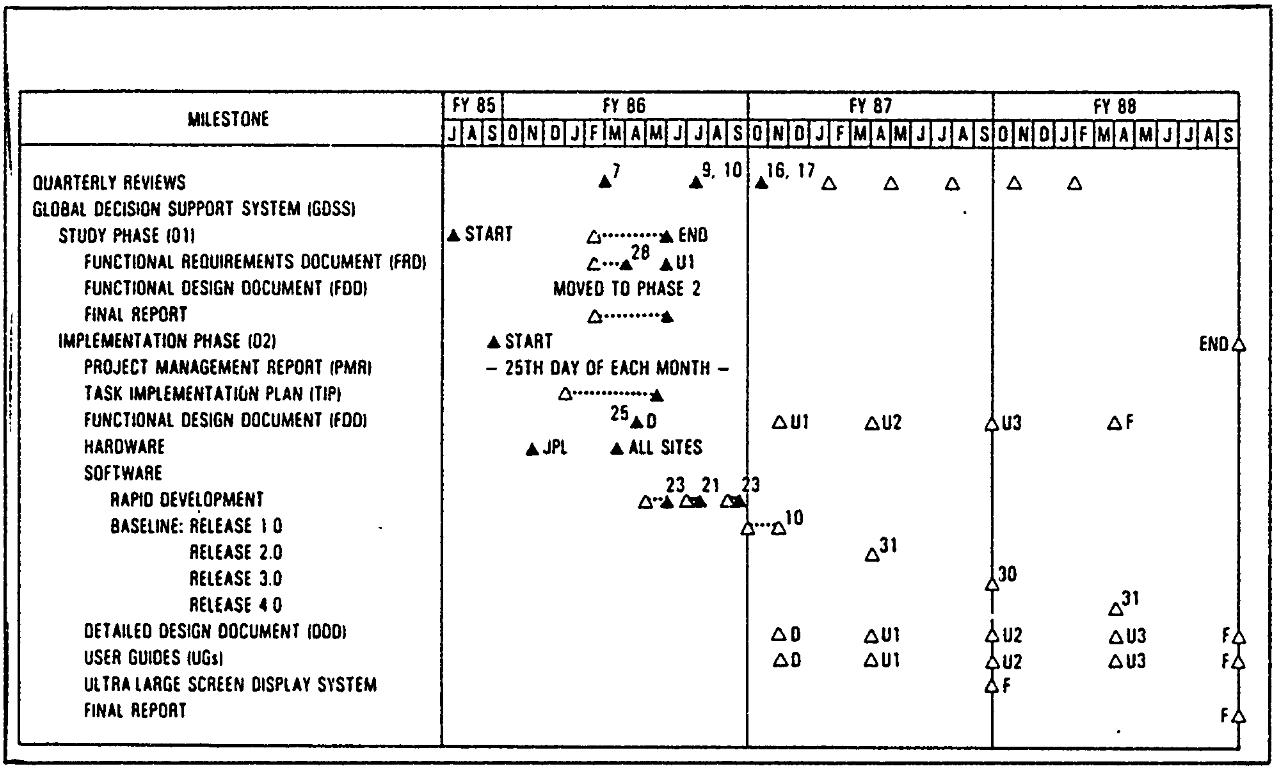

Figure 2
Time Line of GDSS Development
(17:--)

WHY THE PUSH FOR GDSS?

The answer to this question is an important one that cannot be overemphasized. Its roots lie in the basic doctrine of the Air Force. The effort to improve command and control from its current "greaseboard" technology was identified in January of 1985, when the Joint Chiefs of Staff, with the Defense Communications Agency (DCA), selected MAC as the development site for a prototype command center. If successful, the entire DOD command structure will be affected (18:1-1). General Duane Cassidy, Commander in Chief of MAC stated:

> The success or failure of the airlift system depends on three elements: first MAC's people; second, the command and control of people and machines; and third, but not third to any of these, the airplanes and equipment needed to support those airplanes (9:124).

As a result of problems of command and control from the military action in Grenada, the JCS decided to push for an upgraded electronic system (18:4-1). This has resulted in one of the key elements of the GDSS--rapid development to completion in FY 88 with a budget of $16.8 million (16:8). This is not the only reason GDSS is being developed. The basic and overriding rationale needs to be pointed out. This would answer the question, "What is it that we really want to accomplish?" It is important not only to do the job right, but to assure we are doing the right job!

Air Force Manual 1-1 declares the basic doctrine for us to accomplish and capitalize on aerospace power. It lays out the method to determine what job we should be doing. Specifically, fundamental capabilities of aerospace forces must include "responsiveness" as one of its basic combat capabilities. This means that forces must:

> React quickly and bring to bear the force of aerospace power anywhere in the world. The responsiveness of aerospace forces lets commanders employ combat power when hostilities begin. Responsive forces can demonstrate a signal of resolve or intent through deployments, increased stages of alert, or shows of force (14:2-3).

Directly related to responsiveness are the principles of war listed in Air Force Manual 1-1, specifically, the principles of "mass" and "economy of force" (14:2-7). The Air Force does not have enough airlift assets to complete its wartime tasking of 66 million tons/mile/day (9:120). Several proposals, such as the C-17 and C-5B, may alleviate this problem, but even with more airlift assets, poor coordination violates our basic doctrine. It is critical that MAC manage its limited assets in the most economical manner it can to mass forces when called upon. GDSS is an important tool for the commander of airlift forces to determine what the right job is, and how to do that job right.

Chapter Two

WHAT IS ARTIFICIAL INTELLIGENCE?

Artificial intelligence (AI) is "considered to be one of the key technological innovations that will shape our society as a whole in the remaining part of the century" (6:57). Computerization of our society will continue, but will it take a significant leap with the effective use of artificial intelligence? Let's take a common sense view of this complex subject and break it down to its simpler forms. To do this, the discussion will briefly cover the definition of AI, its background, expert systems, and Natural Language Processors. Finally, the chapter ends with a discussion of some of the problems of artificial intelligence.

DEFINITION

One major criticism of AI is that a good definition, universally accepted, simply does not exist. One of the cornerstones of AI is a book by Barr and Fiegenbaum who give this definition:

> Artificial intelligence is that part of computer science concerned with designing intelligent computer systems; that is, computer systems that exhibit the characteristics we associate with intelligence in human behavior--understanding language, learning, reasoning, solving problems, and so on (2:15).

Probably the simplest definition of AI is "the use of the computer to perform tasks that currently require human intelligence" (4:12). There is a big difference between trying to "think" like a human and doing tasks like a human. To identify what AI is in a nutshell--it is very sophisticated software programs, not highly hardware dependent. To support the claim that artificial intelligence is not hardware dependent, a book on artificial intelligence for the Apple line of computers has been published (3:--). The Apple computer is a popular computer, but far from the leading edge of the science! Let's jump right into some of the more important details of artificial intelligence to support the assertion that it is "one of the key technological innovations that will shape our society as a whole in the remaining part of the century" (6:57).

BACKGROUND OF ARTIFICIAL INTELLIGENCE

It is difficult to say if AI is an "evolutionary," or a "revolutionary" change at its present stage of development. One of the major breakthroughs in the science of computers is that the hardware can process "symbols," in addition to numbers. Most computers operate with three major parts: input/output systems (keyboard, printer), central processing unit (CPU), memory (chips, disk drives). First, a program of instructions is loaded into the memory and through the input devices data, that can be recognized and manipulated by the CPU, is sent to an output device. The CPU will process the data through an algorithm. An algorithm is a step by step process used to solve problems. In a hypothetical situation, given a ring of keys and a lock, a computer would start at one end of the ring and try the key in the lock. If it did not work, it would go to the next key until it was successful in opening the lock. Artificial intelligence could also be used to solve this same problem, but would solve the problem in a totally different fashion, though it would still use the same basic hardware structure--here's how.

Artificial intelligence programs are predominately symbolic processors. Instead of following a rigid step by step process outlined above, an AI program would sort through its memory to determine a good "guess" regarding the solution of the problem presented. That is, it uses a "heuristic" approach, or as mathematician George Polya said, "the art of good guessing" (6:37). To clarify using the previous example, an AI program would determine if the lock needed a big key or a little one. Perhaps the program would start with keys matching the name of the lock, and reduce potential misses. This procedure uses the idea of identifying promising approaches to sclving the problem, dealing with partial information, and making an educated guess to the outcome. This "guess" is developed from a knowledge base. The reasoning is based on the rules of thumb loaded with the program software (23:4). As a result, a well developed AI program, can better process large amounts of data. Much of the data base can be ignored by heuristic rules to speed the search for solutions. You might think that a conventional computer could do the same search and come up with the same answer. After all, in keeping with common sense; the right answer, quickly is what we really want. Artificial intelligence goes a step further in getting the right answer--it learns from previous mistakes.

EXPERT SYSTEMS

In a program that uses an algorithm, if a mistake is made, and given the same situation again, the same mistakes will be repeated. An artificial intelligence program "learns" from its previous mistakes and expands on its heuristic knowledge base to avoid the same error. This type of behavior would fit the definition given in the beginning of the chapter for behavior that is normally the domain of intelligence--reasoning and learning. An artificial intelligence program developed to deal with one specific area of knowledge is called an "expert system." Two important parts of an expert system are its knowledge base and a data base of "heuristic" rules. Ideally, an artificial intelligence program would incorporate the knowledge of the world, but storage of such a vast bulk of data is not possible in the foreseeable future. Heuristic rules are general statements of guidance in various situations. Common sense rules of thumb are a good example. Several expert systems are in current use in the U.S. Navy, for example, and are listed in Appendix A. Here is a summary of some dissimilarities between a conventional computer program and an artificial intelligence program:

Conventional	Artificial Intelligence
- numeric process	- symbolic process
- algorithmic approach	- heuristic approach
- correct answers required	- best guess
- hard to change	- easy to change

As stated earlier, AI is mainly a software development. The basic structures of an expert system are:

1. A knowledge base of a [sic] specific facts and heuristic rules associated with the domain.

2. An inference procedure (or control structure) for using the knowledge base. This is the heart of the traditional software programing [similar to an algorithm].

3. A global data base--the working memory of the problem to be resolved (5:71).

AI LANGUAGES

There are many different types of computer languages available for computer programs. This is because some languages are more applicable to the design objective. To illustrate, "BASIC" (Beginner's All Purpose Symbolic Instruction Code) was designed many years ago to be used by people new to computer programing (8:21). Languages specific to AI have also been developed. A language by the name of LISP (List Processing Language) and one named PROLOG (Programing in Logic) were developed to better support the needs of symbolic processing. A word of caution--just because a program is written in one of these languages, does not mean that it will fit any reasonable definition of artificial intelligence. David Warren, one of the original developers of PROLOG, claims:

> The only mistakes you can make are (1) to omit some relevant piece of knowledge from the knowledge base and (2) to include some information which ought not to be there. This will lead to (1) failures to find some of the answers that you intended should be there and (2) getting some unintended answers (6:34).

NATURAL LANGUAGE PROCESSORS

One major goal of artificial intelligence research is to allow human interaction with computers in a natural language (typed, spoken, or printed), as compared to formal computer languages (2:111). The tool being developed to reach this goal is the Natural Language Processor (NLP). Natural Language Processors are a subset of artificial intelligence, just as expert systems are. To operate effectively, the knowledge base of a NLP must have as a minimum the capability to understand:

- The structure of sentences.
- The meanings of words.
- The morphology of words.
- A model of the beliefs of the sender.
- The rules of conversation.
- Extensive shared body of general knowledge about the world (5:111).

In effect, a Natural Language Processor is a type of expert system that interfaces with another, specific type of expert system. What is the present state of the development? Do these types of systems really work? According to William Gevarter, a pioneer in artificial intelligence, "the only systems that performed robustly and efficiently are Type '1' systems--those that do not use explicit world models but depend on key words, or pattern matching, and/or semantic grammars" (2:122). Appendix B lists some of the types and purposes of commercial NLP systems currently in use. Some ways that a Natural Language Processor can be used are:

1. Question & Answering Systems--often used in a limited domain of knowledge and most user inputs are restricted to questions.

2. Natural Language Interface--designed to provide a painless means of communication in the form of instructions or questions to a computer program.

3. Computer Aided Instruction--includes the ability to formulate questions and presentations to the student, and tailor the instruction to his needs.

4. Discourse--systems designed to understand extended dialogue, or speech.

5. Text understanding--very primitive in its development [the ability to read printed material].

6. Text Generation--Demonstrated in the form of spoken words converted to text (5:120).

The development of NLP is faced with many the same problems as other expert systems--namely, the need for a large knowledge base and well defined heuristic rules.

WHAT ARE THE PROBLEMS WITH ARTIFICIAL INTELLIGENCE?

The speed of improvements in artificial intelligence is difficult to predict, but there are several aspects of constructing an expert system which should be kept in mind:

- Few well trained people in artificial intelligence are available.
- Few powerful, well established, and maintained AI languages exist.
- Lack of parallel processor technology.
- Few independent means of checking if conclusions are reasonable.
- Narrow domain of expertise.
- Limited assumptions about problem and solution methods.
- Single expert known as the "knowledge czar."
- Difficulty in the development of the heuristic rules and knowledge base (5:55).

The development of the heuristic rules and the necessary knowledge base is consistently described in the literature as the most difficult part of building an expert system. According to Waterman, a founder of artificial intelligence, the development of heuristic rules may be the most useful by-product from the building of the system (2:23). He proposes that very often organizations do not have their human experts quantify what "rules" they used to do their job. This causes the loss of "corporate memory" when an individual leaves the company. The development of rules is further complicated by the human reaction of not wanting to divulge the source of their "expertise." In one situation, a company expert was embarrassed when an interviewer found his job duties were completely described by a few guidelines (5:35). The employee was disconcerted that he could so easily be replaced with a few well defined rules. The encoding of the rules of thumb requires an experienced artificial intelligence programmer to ask the right questions.

Chapter Three

SHOULD ARTIFICIAL INTELLIGENCE BE INTEGRATED WITH GDSS?

The aim of this chapter is to answer the question: Should artificial intelligence be integrated into the Global Decision Support System? To accomplish this aim, this chapter will discuss the basis for change of the system, the time value of data, an adaptable interface with GDSS, conversion of GDSS software to a common computer language--ADA, "backdoors", and "worms." Integration of AI with GDSS does have its problems and these specific obstacles discussed include: the adaptive interface challenge, technical dilemmas, and organizational problems. To begin, what is the reason to change a system already designed and being installed?

THE BASIS FOR CHANGE

As stated in Chapter 1, GDSS was originally tasked to:

-- Act as an evolutionary step in the overall command and control upgrade program.
-- Provide the DOD with a prototype command center system (18:1-1).

This original tasking is the basis for the proposed integration of artificial intelligence with GDSS. Further, it suggests the importance of making the system compatible throughout the services. Implied, but not specifically stated, is that the GDSS network should enhance the operational capability of MAC. The program goals give the clear indication that improvements should be incorporated when available and barriers to change are contrary to overall program objectives. Just how could an artificial intelligence based expert system be used within GDSS? First, the concept of time value of data needs to be understood.

TIME VALUE OF DATA

When two pieces of information arrive at a computer, the importance of each piece of information is arbitrarily based on which one arrived first. For example, if a routine summary of an operation is received by a computer and then, a moment later, information of higher interest to the operator is transmitted, the conflict is resolved by the network operating system. This is known as collision detection/collision avoidance. Each packet of information would be processed in turn as it was received based on time. Common sense would be to overview each packet to determine if it was urgent or routine in nature, and bring to the operator's attention items of importance. A few rules of thumb could be:

- -- Is the message listed as routine, priority, or urgent?
- -- Does the message refer to exercises or code words that are being closely watched?
- -- Does the message refer to aircraft damage or injuries?

Other rules of this nature are not difficult to develop. Not only is it important to prioritize the data, but to maximize its value. Normal computer networks do nothing to prioritize the value of the data received.

A big problem in command and control is the task of sorting out the brush fires from the forest fires. As data supplied to decision makers increases, the need to separate and prioritize the information becomes critical. This is the time value of data; meaning that as information is received, the value or usefulness of the data may be very high, but as time goes on the ability to react to it successfully diminishes. For example, a command post controller learns of an aircraft fire in progress. If the situation had just occurred, he could successfully limit collateral damage with quick reactions. If, however, the fire occurred several hours ago, the value of the information is "overcome by events." In short, the value of information is high at first, but loses its utility as time goes on. "The British are coming!" was informative in 1776, but has less critical time value today. AI can help sort out data and priortize it.

The very name of GDSS focuses on two areas. First, it is a "global decision" system specifically created to help commanders maintain situational awareness, or see the global picture. Second, GDSS is intended as a "support system" to the decision makers, not simply by increasing the amount of data supplied to them, but rather by supporting them through priortizing information. The object is to give the commander the information he needs quickly; not just more data. A well developed artificial intelligence program can prioritize incoming data and there by optimize or take advantage of the time value of data concept. This type of expert system is to be used to support the commander, and not as an interface between every console unit within the system. In effect, the expert system should monitor the GDSS network; not run it.

ADAPTABLE INTERFACE WITH GDSS

The reason GDSS was proposed in the first place was because of command and control problems following the Grenada operation. This operation was an example of the importance of the time value of data. The capability to connect to GDSS's Wide Area Network to supply real time information to the other system users needs to be developed. For example, if the Grenada operation were repeated, it would be advantageous to deploy a mobile computer capable of connecting to the GDSS via AFSATCOM or telephone lines. Furthermore, it would need to be adaptable to various other communication mediums. One such unit could be place at Pope AFB and another on the island of Grenada itself. Deployed computers would feed the rest of the system but not be a mirror of the entire system. Thus, when an aircraft departs from Pope AFB the network is updated with the change reflected at the other GDSS nodes including Grenada. The key point is that the system can be deployed quickly and is rugged enough to use in the field. This concept was proven by the 60th Military Airlift Wing at Travis AFB.

During several operations (ORI's and ORE's), Travis AFB deployed simple Zenith computers with the Air Lift Control Element (ALCE). When people were processed through the mobility processing line at Travis AFB, their AFSC, name, rank, etc., was recorded. This computer information was then transmitted directly to the ALCE unit deployed. In almost all cases, the ALCE knew exactly who was on the aircraft inbound to them before the aircraft had even departed Travis AFB! Thus, the ALCE, and wing commander, knew who was coming and could immediately put them to use on the right job, without delay after they arrived. The process was simple and effective. Each commander had a sorted list of who was on station, including a list of aircraft movements that had occurred, and were scheduled to occur. Events could be anticipated instead of reacted to. This need is important in Europe as well.

The Russians have been kind enough to tell us through their military doctrine that they will target our command and control centers in the event of hostilities (11:10). It is imperative that we anticipate the destruction of the GDSS node at Ramstein AB, for example, and compensate for it. This would involve a more sophisticated version of an adaptable interface with GDSS.

Clearly, in a theater level conflict in Europe, the battle lines will be fluid and nonlinear; the physical location of Ramstein makes it vulnerable. We need the capability to support an operation from other secure locations. The more fluid the operation, the greater the need for a user-friendly interface. Therefore, the author recommends that a backup system be developed to replace a node lost from terrorism or attack. Additionally, the project should obtain microcomputers capable of interfacing the GDSS network from any location. These computers would have to connect to the system in a secure, yet adaptable way. The loss a major section of the WAN must be anticipated and planned for. Command and control is not management, it is leadership. The reason GDSS is being developed is not to demonstrate a computerized operation, but rather improve airlift effectiveness in peace and war. Finally, the people and equipment need to be exercised to identify shortfalls. A well written AI program could help pull all of this together.

CONVERSION OF GDSS SOFTWARE PROGRAMS TO ADA

There are two more areas which could strengthen the Global Decision Support System. First, the conversion of the software programs to a common language--ADA. The Department of Defense decided to standardize all programs written for its use by selecting ADA as the standard computer language (7:3). There are advantages and disadvantages to this decision, but significant improvements in the portability and standardization across the services can be realized. Further, the conversion to ADA would have a great effect on improving the maintainability of the overall system. However, the Military Airlift Command should make no attempt to organically develop this type of expert system. Instead, a wait and see policy should be adopted to take advantage of civilian developments or research by the Department of Defense in the conversion of computer languages. The old adage--if it works, don't fix it--is surely true, but as time goes on, changes to the GDSS software will be requested. It is immensely difficult to change a program that is not standardized with the rest of the system. One small change can have the most fiendish effect on another section of the program. This must be further qualified, since the conversion should <u>not</u> be made until sufficient reliable ADA compilers are developed. Here is why.

A compiler is a machine specific program that converts the software program typed by a human into a form that can be used by a computer directly (8:479). What this means is that without being compiled, a computer must refer to its own instruction set on how to behave for each "word" in the program before performing it. A compiled program eliminates this time consuming step-by-step process, and allows the program to be executed immediately since it is already in that specific machine's language. As described in Chapter 1, the software code of GDSS is written in several different computer languages. For example, JPL decided the Command Post functions would be written in FORTRAN. The decision was logical, even though the Department of Defense guidance is that projects, such as this, will be written in ADA (7:3). The reasons behind this decision were that few experienced ADA programmers were available, and the accelerated development schedule of the program (20:--). Secondly, and very importantly, there are no suitable ADA software compilers that will convert the high level code into DEC MicroVax executable code. Thus, the speed of the program is severely limited by the fact that each instruction must be interpreted by the DEC minicomputer before it can be executed. The compilers simply do not exist at this time. The second application of artificial intelligence to GDSS software code is for "backdoors" and the fatal embrace of a "worm!"

BACKDOORS

A backdoor is any method of accessing the data base of a computer system by a code word that is outside of the normal procedure (1:29). It bypasses the normal log-on procedures and security protection. The movie "War Games" is a good example where the computer designer used his son's name as the means to access the inner workings of that particular expert system. Most programmers will include some sort of personal identification or access into their programs when they are writing them. This allows them to quickly access the main program and avoid tedious log-on procedures. However, this is not desirable in a system that controls military airlift forces and which later could be the basis for DOD command and control.

WORMS

A worm is simply part of a computer program that when executed will destroy all, or part of the computer's data base. A noteworthy example was when a bank worker programed a computer to erase the bank's records if his name no longer appeared on the payroll (1:30). Thus, when he was fired from his job, at the end of the month when his name was no longer on the payroll, all bank data was erased. This type of code is not difficult to do. For example, in BASIC computer language the following line would erase all computer files every January 1st.

```
IF VAL(DATE$)=1 THEN DELETE *.*
```

In another example, the worm below would lie dormant until the "SIOP" plan was implemented and then erase all program and data files including itself!

```
IF PLAN$="SIOP" THEN PRINT "One Moment Please": delete *.*
```

It is not difficult to construct a worm to destroy a system if a specific operational plan is implemented. All that is required is the name of the specific plan and how it would be loaded into the computer. The real danger of worms is that they are very easy to hide within a long computer program. The previous examples only took one line of code and yet, are rather nasty in their outcome. It is also possible to cause the computer to transmit its data via telephone lines when a specific plan is activated, and thereby giving away our force structure. This worm would take a little more work, but is well within the range of the possible.

Just as it makes sense to look inside the electronic equipment of items placed in controlled areas to make sure that there has been no eavesdropping devices or transmitters placed inside, it makes sense to look inside the software programs to make sure that there are no surprises awaiting execution. For instance, an expert system could focus on all commands that called for input and output of data, particularly the erasure of files. It would not even have to be 100% correct in its evaluation because the expert system could simply identify suspect lines of code, and then a human programmer could examine the code more closely. If all the programs of GDSS were converted to ADA, this task would only require one such expert system, otherwise, an expert system knowledgeable in FORTRAN, COBAL, ADA, etc would be necessary. The sheer size of the computer programs in the Global Decision Support System makes it easy to hid a worm.

This section is not meant by any means to impinge on the veracity or honesty of the JPL programmers. There is every reason to believe that well after the programs are delivered, a disgruntled worker could modify the code. This implies that during any phase of development and use of the program, illegal modifications might be made. Thus, examination of the code should be done regularly throughout the lifetime of the program as part of its normal maintenance. Admittedly, this would have only a small deterrent effect as it is difficult to determine who is the guilty party, but at least the problem would be identified. The program could also be used to detect all passwords that could be used as a backdoor and thus prevent unauthorized entry to classified levels of data. This type of expert system could be used on other programs written in ADA throughout the DOD.

SPECIFIC RECOMMENDATIONS

To pull everything together, the author recommends that artificial intelligence be integrated with the Global Decision Support System, but NOT attempt this action until several specific milestones have been achieved. It is true, AI is on the verge of tremendous accomplishments in the areas of expert systems, however, this recommendation is made because while integration with GDSS is technically possible, it is operationally impractical until certain prerequisites can be completed. Two problem areas are addressed; technical and organizational. First, the technical problems and its associated milestones.

TECHNICAL PROBLEMS

Although great strides have been achieved in limited applications of AI, a review of the current state-of-the-art finds that "most current applications [sic] work involves very few substantive new ideas over earlier systems" (12:65). A study of the literature, such as in the bibliography, will frequently reference two or three specific operating and highly effective AI systems, but little more practical applications have been accomplished to date. The technical bottle necks are too large to successfully integrate an expert system to GDSS at this time. As an example, the development of the "knowledge base" is still in a very basic stage of growth (13:59). As discussed earlier, it is very cumbersome, time consuming,and dependent on the skill of the programmer. More experience must be gained at obtaining this type of knowledge by learning to ask the right questions before the implementation of an AI system. To further cloud the maturatioin of an expert system, the computer languages for AI have not yet stabilized, and are under constant change as the market develops. Of the systems currently in use, their domain of knowledge is extremely narrow, and have not been expanded significantly. Furthermore, just because a program is written in LISP or PROLOG does not mean that the program is capable of artificial intelligence. In short, AI is still in the research stage instead of the operational stage.

A second technical problem is speed. The speed of an AI computer system is hampered by several hardware problems. The introduction of 32 bit processors, such as the Intell 80386 microprocessor, will greatly help the situation, but what is really needed is for parallel processor technology to emerge. Parallel processing means that a computer can perform more than one function simultaneously. This is the same as the human body where we regulate body temperature and at the same time perform higher level functions. This is a significant milestone. In addition, the equipment purchased needs to be of the garden variety, or "off-the-shelf." Murphy's law would predict that failures would occur in the system. A one of a kind computer, just perfect for AI, could be difficult to maintain. It is much smarter to repair, or replace a computer with existing hardware, on location, instead of hoping for contractor support.

The conversion of the software code to ADA will be time consuming and fraught with numerous bugs. It is a necessary step in the overall evolution of any system for the DOD. It is pointless at this time to convert the other GDSS programs without the necessary compilers to change the source code into executable code. Once converted to ADA, the problem of "backdoors" and "worms" will never go away as a threat. As a minimum, the source code should be independently verified, compiled, and archived. The program length can be compared to the current software in use to insure that no "extra" lines of programming have been added. The stakes are too high to just hope for the best. Later as other expert systems are developed to accomplish these measures, they should be used regularly. It is cost prohibitive, in terms of time and money, to attempt such an undertaking at the present level of technology. Opportunity costs could be excessive.

Natural Language Interfaces with expert systems are at a very crude level. It does little good to have a well developed system if it is cumbersome and difficult to get the information that is needed out of it. One thing that is not needed is another computer system for users to learn additional command sets of instructions. Make it friendly, make it usable, make it work for you, instead of the other way around. Natural Language Interfaces are nothing more than an expert system in themselves. As other areas of expert systems develop, so will this type of interface. However, a Natural Language Processor should not be a substitute for basic computer training. It would be a waste of time and effort to develop a NLP, when the real need is a computer literate user. These have been the technical problems; now for the organizational ones.

ORGANIZATIONAL PROBLEMS

Artificial intelligence programmers are too scarce, expensive, and generally inexperienced (13:59). Recently, PROLOG and LISP computer languages have been manufactured for microcomputers. As more computer users are exposed to these languages, a better and larger programmer base will emerge. There are several systems under development that use an expert system to write the software for another expert system. Never the less, it is doubtful that in the very near future, a fully developed system, such as needed for GDSS, will be available. If successful, such a system could relieve the programmer of some of the more tedious requirements of writing the program; significant time, and therefore, money savings will be realized.

The experience level of Air Force "blue suiters" in artificial intelligence is scant to say the least. As can be seen by Appendix A, the Navy has several artificial intelligence systems already on the books. These beginnings are necessary for future developments in artificial intelligence. But, are these programs really capable of artificial intelligence, or are they simply well developed software programs with extensive default values that just happen to be written in LISP or PROLOG? Do they, in fact, conform to generally accepted definitions of AI, or did the programmers simply name them so? Without a detailed analysis, no determination can be made. It would be appropriate at this time to develop a cadre of Air Force personnel to become familiar with the progress of other services and prepare for implementation when the state-of-the-art reaches the milestones presented. Why develop this organizational resource?

The success or failure of nearly any system is dependent on the quality of the people involved. This has been true throughout history. The development of "blue suit" cadre is essential since presently a large gap exists between the developers of a high technology system and the users of the system. The larger this gap, the more likely mistakes will be made in its application to real world problems. AI will, more and more, be hailed as the "solution" to many problems. Good people, educated in the technology of AI, can keep programs aimed at the right objectives. The right people and good leadership will make the system work, not technology alone.

CONCLUSION

The bottom line is this--to implement an artificial intelligence expert system with a Natural Language Interface at the present state-of-the-art is a mistake. However, action should be taken to increase the awareness of breakthroughs by development of a core of "blue suit" experts in the area to capitalize on their knowledge. Many young developers of artificial intelligence technology are more likely to get caught up in the research of the subject, for its own sake, and lose sight of the objectives and importance of a real world application. We cannot afford big promises and few actual accomplishments. To counter this tendency, the author recommends development of a cadre of Air Force experts capable of monitoring the construction of an AI system. Coupled with the growth of a trained cadre, education of the end-users in how to effectively implement the system is essential. Specific milestones are improvements in the knowledge base representation, availability of experienced programmers, speed/parallel processor breakthroughs, use of off-the-shelf hardware, and improvements in Natural Language Interfaces to allow for truly user friendly interaction balanced with user literacy. Further, the software code should be examined for backdoors and worms by human programmers until an appropriate expert system can be developed.

A critical need right now is an interface with the GDSS network capable of being deployed to support any operation throughout the spectrum of conflict, from low intensity conflict to theater warfare. Project Forecast II specifically identified artificial intelligence as one of the initiatives that "will revolutionize the way the Air Force carries out its missions in the 21st Century" (15:--). The report went on further to state "the potential to share hardware and software in a variety of systems under an artificial intelligence 'manager' will result in significant improvements in reliability, maintainability, and supportability" (15:--). The time to lay the foundation is now.

BIBLIOGRAPHY

A. REFERENCES CITED

Books

1. Baker, Richard H. The Computer Security Handbook, Blue Ridge Summit, PA: TAB Professional and Reference Books, 1985.

2. Barr, A. and E. A. Feigenbaum (eds.). The Handbook of Artificial Intelligence, Vol 1. Los Altos, Calif: William Kaufman, 1981.

3. Forsyth, Richard and Chris Naylor. The Hitch-Hiker's Guide to Artificial Intelligence. Applesoft Basic Version. New York: Chapman and Hall/Methuen, 1985.

4. Gevarter, William B. Intelligent Machines: An Introductory Perspective of AI and Robotics. Englewood Cliffs, NJ: Prentice Hall, 1985.

5. Hayes-Roth, Fredrick, Donald A. Waterman and Douglas B. Lenat. Building Expert Systems. Reading MA: Addison-Wesley Publishing Co., 1983.

6. Hayes, J.E. and D. Michie. Intelligent Systems: The Unprecedented Opportunity. New York: Ellis Horwood Ltd., 1985.

7. Price, David. Introduction to ADA. Englewoods Cliffs, NJ: Prentice-Hall Inc., 1984.

8. Seidman, Arthur H. Editor. The Handbook of Computers and Computing. New York: Pratt Institute, Van Nostrand Reinhold Co., 1984.

Articles and Periodicals

9. Cassidy, Duane, General, Commander in Chief Military Airlift Command. "MAC's Moment of Truth." Air Force Magazine, September 1986, pp. 115-131.

CONTINUED

10. Correll, John T, Editor in Chief. "Machines That Think." Air Force Magazine, July 1986, pp. 70-75.

11. Dues, Mark L. "The Soviet Strategic Offensive: Controlling the Level and Duration of Conflict." Theather Warfare, Vol 9. January 87. Air Command and Staff College. Maxwell Air Force Base, AL., pp 5-24.

12. Hoffman, Paul E. "Expert Systems 85: Proceedings of the Fifth Technical Conference of the British Computer Society Specialist Group on Expert Systems." BYTE: The Small Systems Journal, Vol 11, No.12 (November 1986), pp. 65-70.

13. Rose, Kenneth H., Major, U.S. Army. "Why Artificial Intelligence Won't Work." Military Review: The Professional Journal of the US Army. (December 1986). Ft. Leavenworth, Kansas: US Army Command and General Staff College, pp. 57-63.

Official Documents

14. Air Force Manual 1-1, "Basic Aerospace Doctrine of the United States Air Force." 16 March 84.

15. Project Forecast II, Executive Summary. Headquarters, Air Force Systems Command, Andrews Air Force Base, MD.

Unpublished Materials

16. Command and Control Automation Project. Global Decision Support System (GDSS). "Defense Information Systems Program Functional Requirements Document", 27 May 1986. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

17. Command and Control Automation Project. Global Decision Support System (GDSS), Quarterly Review. October 17, 1986. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

CONTINUED

18. Global Decision Support System (GDSS). Task Command and Control Automation Project Task Implementation Plan. 24 April 1986. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

19. Moran, J.F. A Concept for System Management of Software for Command and Control Systems. Maxwell AFB, AL: Air University. June 1965.

Other Sources

20. Poff, Richard, Lt Col, USAF. Hq MAC/DOA, Scott Air Force Base, IL. Interview, 17 Oct 1986.

B. RELATED SOURCES

Books

21. Clifton, Pat O., Col, USAF AI : A User Friendly Introduction. No 85-1. Maxwell AFB, AL: Center for Aerospace Doctrine, Research and Education, Air University Press. March 1985.

22. Fritz, James S., James Charles, F. Kaldenbach and Louis M. Progar. Local Area Networks--Selection Guidelines. Englewoods Cliffs, NJ: Information Systems & Network Corp., Prentice-Hall, Inc., 1985.

23. Gevarter, William B. Artificial Intelligence Expert Systems Computer Vision and Natural Language Processing. Office of Aeronautics and Space Technology National Aeronautics and Space Administration, Washington D.C. Park Ridge, NJ: Noyes Publications, 1984.

24. Haugeland, John. Artificial Intelligence: The Very Idea. Cambridge, Massachusetts: A Bradford Book: The MIT Press, 1985.

25. Saram, Hugh de. Progamming in Micro-PROLOG. New York: Ellis Horwood Ltd, 1986.

CONTINUED

26. Simons, G.L. Introducing Artificial Intelligence. NJ: Manchester Inc., 1984.

Articles and Periodicals

27. Kimbrell, Roy E. "English Recognition: The Ultimate in User Friendliness." BYTE: The Small Systems Journal (December 1985 Vol 10 No 13), pp. 125-140.

Unplublished Materials

28. Command and Control Automation Project. Distributed Management Information and Control System (DMICS), Quarterly Review. October 16, 1986. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

29. Deforest, Lloyd. Distributed Management Information and Control System (DMICS): Sustaining Engineering Work Item Analysis Status Document. Status as of 13 Oct 1986. Pasadena, CA: Jet Propulsion Laboratory, California Institute of Technology.

APPENDIX

APPENDIX A

Expert Systems Currently in Use
(with heavy reference to military applications)
Source (5:164-176)

System	Purpose (Comments)
DEVISER	General Purpose Automated Planner/Schedular to generate plans to achieve goals with a time constraint. Schedule space craft actions during planetary flybys. (Jet Propulsion Laboratory).
KNOBS	Planning consultant for USAF tactical missions. Other uses: - Naval "show of flag" missions -Scheduling of crew activities for the NASA space shuttle. (Uses a Natural Language Interface, used to assist use by accepting mission data and devises a solution to the user's problem while checking for inconsistencies and oversights. Developed by the MITRE Corp.).
ISIS-II	Job-shop Planning/ scheduling of Parts Production. Generates schedules by heuristic search using evaluation functions based on constraints of costs, process applicability, machine availability, and supervisor preferences.
BATTLE	Battle field weapons assignment. Acts as a consultant to assign the proper weapon to a specific target. (Developed by the NRL AI Laboratory) (23:82).

CONTINUED

AIRPLAN	Naval Aircraft Operations. Used as a flow planner for aircraft operations from naval carriers (Developed by CMU) (23:80).
TECH	Naval Task Force Threat Analysis (RAND/NOSC) (23:80).
ANALYSIST	Military Situation Determination (Developed by the MITRE Corp).

APPENDIX

APPENDIX B

List of Commercial Natural Language Processor Systems
Source (23:121)

System	Purpose (Comments)
INTELLECT	NLI for data base retrieval (Several hundred sold; takes about 2 weeks to implement for a new data base).
PEARL	Custom NLI (Large start-up cost in building the knowledge base; several systems sold).
Straight Talk	Highly portable NLI of data base management systems for microcomputers. (Very compact and efficient; written in PASCAL).
SAVVY	System interface for microcomputers (Uses adaptive or best fit pattern matching).
ALPS	Interactive Natural Language Translation (uses a dictionary that provides various translations for technical words. User selects among displayed meanings).