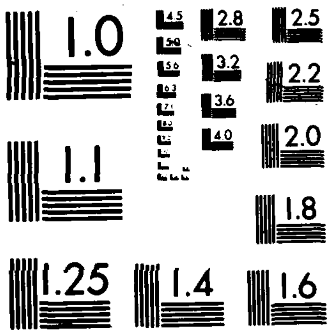
AD-A175 456 AN ASSESSMENT OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TECHNOLOGY FO (U) HARRY G ARMSTRONG AEROSPACE MEDICAL RESEARCH LAB WRIGHT-PATTE W L MARTIN SEP 86 1/2

UNCLASSIFIED AAMRL-TR-86-040 F/G 6/4 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3

12

AAMRL-TR-86-040

AD-A175 456

AN ASSESSMENT OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TECHNOLOGY FOR APPLICATION TO THE MANAGEMENT OF COCKPIT SYSTEMS (U)

WAYNE LEE MARTIN

ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY

SEPTEMBER 1986

Approved for public release; distribution is unlimited.

HARRY G. ARMSTRONG AEROSPACE MEDICAL RESEARCH LABORATORY
AEROSPACE MEDICAL DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573

NOTICES

TECHNICAL REVIEW AND APPROVAL

AAMRL-TR-86-040

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

CHARLES BATES, JR.
Director, Human Engineering Division
Armstrong Aerospace Medical Research Laboratory

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A175456

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
AAMRL-TR-86-040

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION
Harry G. Armstrong Aerospace Medical Research Laboratory

6b OFFICE SYMBOL *(If applicable)*
AAMRL/HEA

7a NAME OF MONITORING ORGANIZATION

6c ADDRESS *(City, State, and ZIP Code)*
Wright-Patterson AFB OH 45433-6573

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.
62202F	7184	11	45

11 TITLE *(Include Security Classification)* AN ASSESSMENT OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TECHNOLOGY FOR APPLICATION TO THE MANAGEMENT OF COCKPIT SYSTEMS (U)

12 PERSONAL AUTHOR(S)
Martin, Wayne L.

13a TYPE OF REPORT
Technical

13b TIME COVERED
FROM ______ TO ______

14 DATE OF REPORT *(Year, Month, Day)*
1986 September

15 PAGE COUNT
126

16 SUPPLEMENTARY NOTATION

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Artificial Intelligence
Expert Systems Technology
Cockpit Systems

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

A review of the literature in the field of artificial intelligence was performed to identify research and development efforts in industry, academia, and government laboratories that may be related (or relatable) to the cockpit management function in tomorrow's aircraft.

Individual chapters address the following topics: Chapter 1 - An Introduction to Artificial Intelligence and Expert Systems; Chapter 2 - Artificial Intelligence Development Applications in DARPA, DOD, and NASA; Chapter 3 - State-of-the-Art Review and Projection of Future Expert System Developments; Chapter 4 - Human Factors Research in Artificial Intelligence and Expert Systems; Chapter 5 - Image Understanding; Chapter 6 - Natural Language Processing/Understanding, and; Chapter 7 - Summary Comments on the Development and Application of Artificial Intelligence and Expert Systems.

Separate bibliographies are provided at the end of each chapter to assist the reader in identifying specific literature of interest. A glossary of abbreviations, acronyms, and special terms used in the context of this report is also provided.

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

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
Wayne L. Martin

22b TELEPHONE *(Include Area Code)*
(513) 255-7603

22c OFFICE SYMBOL
AAMRL/HEA

Summary

Pilot workload in the cockpits of modern tactical aircraft has grown in proportion to technology advancements in sensor, display, and electronic capabilities, as well as ever increasing mission requirements in the interests of versatility and survivability. These combined influences have created a serious problem of information overload for today's pilot and, unabated, will likely grow worse in the future. Even a cursory analysis of the growth of the number of controls and displays throughout the history of military (and civilian) aircraft testifies to the exponential growth in information processing overhead we are continuing to place on our pilots.

The objective of this effort is to review the literature in the area of artificial intelligence (AI) to identify research activities and development efforts that may be related (or relatable) to the management of aircraft systems by the pilot.

Since the role of the pilot has shifted through the years from that of an airframe controller to a manager of complex systems, the initial chapter of this report provides a survey of the literature that includes the application of AI in general, and expert systems in particular, to management problems across a wide range of disciplines.

Chapter 2 provides an overview of ongoing and planned efforts by various government agencies to develop a broad range of AI systems, supporting both military and civilian applications.

Chapter 3 picks up where Chapter 2 leaves off to discuss expert system developments within industry (many of them directly supporting, or in preparation to support, government projects) together with an indication of future directions of the technology and discussion of issues and concerns voiced by leading experts in the field. Since the material covered in Chapter 3 is of considerable breadth, a separate summary is provided as an overview of the technology, applications,

problems, successes and research and development needs that were identified.

Chapter 4 treats what has been deemed by many AI practitioners as the most important design aspect of expert systems, the interface between the system and the user. Several of the more prominent design considerations that have received research attention are discussed, together with an indication of the cooperative research and development activities that need to be performed jointly by human factors and expert systems specialists to assure the interface problem is adequately addressed.

Automated image understanding is the subject of Chapter 5. Although this is a technically difficult area, significant progress is being made on both the scene analytic and processing hardware technology fronts that promise at least potential breakthroughs in the decades to come.

Chapter 6 reviews the technology and research directions in the processing and understanding of spoken natural language. Although some of the problems in this area are technically similar to those in visual image understanding, speaking with brief pauses between words and using constricted vocabularies permit useful applications of speech recognition systems today.

In each of the above chapters, parallels between the research and development activities being reviewed and actual or projected pilot cockpit management functions are drawn where appropriate.

The final chapter draws from all previous material to summarize projections, cautions and promises associated with the development and application of AI and expert systems. Readers wishing to acquire a better appreciation for the content of the total report, without having to read each individual chapter, might do well to review the summary section of Chapter 3 (pp 59) and then read Chapter 7.

PREFACE

This work was accomplished largely while the author was on long-term, full-time training (at Wright State University, Dayton OH) under the sponsorship of the Armstrong Aerospace Medical Research Laboratory (AAMRL), during the period September 1985 to June 1986. Efforts expended before and after that period were performed under the Human Engineering (HE) Division Project 7184, Task 11, Work Unit 45. The report provides the basis for a dissertation submitted to Columbia Pacific University in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Engineering Psychology).

Since this author wished to conduct as thorough a review of the relevant artificial intelligence (AI) literature as was reasonably possible, and at the outset had few preconceived notions as to the specific applications of AI that might be encountered in the literature, a multifaceted approach to the literature review was developed. This included computerized keyword searches on titles and abstracts in the open literature, titles and abstracts of technical reports and work units in DTIC (Defense Technical Information Center), manual searches of citation indices, the holdings of three university libraries and three Air Force libraries, and each new book or periodical that was found to be released on the subject over approximately the September 1985 - June 1986 period, together with many technical discussions with local (Air Force and university) personnel knowledgeable in various aspects of the AI field.

Because the materials discussed in each chapter of this document are largely independent from that treated in other sections (with some possible overlap between Chapters 2 and 3), a separate biliography is provided at the end of each chapter. To facilitate an overall perspective of the total coverage, as well as provide quick access to specific topics, a Master Table of Contents is provided at the beginning of the document.

I wish to thank Dr. Thomas A. Furness, Chief of the Visual Display Systems Branch, and Mr. Charles Bates, Jr., Director of the Human Engineering Division for their strong support and encouragement throughout this project.

Special thanks go to Dr. Herbert A. Colle, Chairman of the Department of Psychology at Wright State University who expertly reviewed draft sections of this volume while the author was under his tutelage in a series of independent reading courses at Wright State. His comments and suggestions were always helpful and stimulated deeper probes into many areas than would have otherwise occurred, resulting in numerous quality improvements to this report.

I am indebted to Mrs. Sandra A. Stevenson of the Technology Integration Branch of the AAMRL for her final editorial review of the report.

Deepest appreciation is also extended to the person responsible for the extraordinarily efficient and conscientious typing of this manuscript, Miss Tanya S. Ellifritt of the Visual Display Systems Branch of the AAMRL.

MASTER TABLE OF CONTENTS

CHAPTER 1 - An Introduction to Artificial Intelligence and Expert Systems 9

Introduction 9
A Survey of the Literature 10
Some Application Examples 14
Projection of Expert Systems Technology into the Cockpit Management Arena 17
Conclusions 21
Bibliography 22

CHAPTER 2 - Artificial Intelligence Development Applications in DARPA, DOD, and NASA 24

Introduction 24
DARPA Sponsored Efforts 25
NASA Sponsored Efforts 28
Air Force Sponsored Efforts 31
Navy Applications 34
Army Sponsored Efforts 35
Summary 36
Bibliography 36

CHAPTER 3 - State-of-the-Art Review and Projection of Future Expert System Developments 39

Introduction 39
Industrial Expert System Developments 40

Forecasts of Expert System Technology 45
Basic Research and Philosophical Issues 52
Rational Skepticism of AI Claims 55
Lessons Learned 57
Summary 59
    The Technology 60
    Applications 61
    Problems 62
    Successes 64
    Primary Needs 65
Bibliography 65

CHAPTER 4 - Human Factors Research in Artificial Intelligence and Expert Systems 68

Introduction 68
Design Considerations for Expert Systems 68
The System and Users Need Models of Each Other 71
Evaluation of User Performance 74
Summary 78
Bibliography 79

CHAPTER 5 - Image Understanding 82

Introduction 82
Processing Architecture 83
Model Developments 84
Some Applications 89
Conclusion 93
Bibliography 94

CHAPTER 6 - Natural Language Processing/Understanding 97

Introduction 97
Speech Recognition and Coding 97
Understanding Natural Language 102
Some Applications Research 107
Summary 112
Bibliography 113

CHAPTER 7 - Summary Comments on the Development and Application of Artificial Intelligence and Expert Systems 116

Bibliography 120

Glossary of Abbreviations, Acronyms, and Special Terms 121

CHAPTER 1

An Introduction to Artificial Intelligence and Expert Systems

Introduction

This chapter provides an overview of the current literature in the general field of artificial intelligence (AI), together with an exposition on the subarea of expert systems. Some examples of prominent applications of expert systems in a variety of disciplines are provided to generate a framework for understanding how such systems could and are being developed to support cockpit management problems, as well as other military and civilian applications.

The growth rate of expert systems will be shown to be as dramatic as their potential for changing the way in which functions across a broad spectrum of disciplines may be performed in the future. We are at the threshold of a new science in the exchange of information, knowledge, and exploration of reasoning processes.

Artificial intelligence is a scientific/engineering discipline that can be broadly defined as the application of computers and programs to tasks that normally would require reasoning and perception by humans. Within this discipline there is a broad range of interests and activities, including the subareas known as expert systems, planning and problem solving, robotics, computer vision/image understanding, and natural language communication, as well as explorations into the nature of intelligence itself. It is the subarea of expert systems and its supporting field of knowledge engineering that seem to provide the greatest near-term potential for application to cockpit management problems. Expert systems are meant to provide the user with a sufficient set of rules, facts, characteristics, and conditions so as to reflect the reasoning processes and expertise of a knowledge domain specialist (e.g., a flight control expert, or a master mission planner) all of which may be brought to bear on the problem at hand. Applications of these systems are being developed across a broad range of technical areas. Manuel (1985) reports that nearly every major U.S. corporation either has or is initiating an AI group to develop or explore expert systems in their technology areas.

Although the earliest efforts were performed mostly by computer scientists, major contributions are now being made across a broad multidisciplinary spectrum. For example, there are engineers concentrating on robotics and image analysis/pattern recognition; operations research people are developing search techniques and optimal decisions under uncertainty; cognitive psychologists are studying human systems for pattern recognition and semantic analyses; philosophers and linguists are attempting to synthesize intelligent behavior for application to natural language understanding systems, while persons across the spectrum of disciplines are developing expert systems (albeit most to only the demonstration prototype phase) to mimic the application of their knowledge in some constrained problem area.

A Survey of the Literature

The first efforts made to review the AI literature were computerized searches of listings in the Engineering Index and of technical report titles and work units within DTIC (Defense Technical Information Center). Although these searches identified some relevant and useful documents, their content proved to be relatively disappointing for the purposes of this effort. Upon surveying other available citation indices, two prominent periodical literature reference sources were identified (i.e., the Applied Science and Technology Index and the Business Periodicals Index) that contained the breadth and depth of citations needed to initially gain an appreciation for the extent of existing literature in the area. A review of the Applied Science and Technology Index, starting with 1958, indicated that the first entry for

AI occurred in 1966. The article was entitled simply "Artificial Intelligence", and was published in Scientific American by MIT Professor M.L. Minsky (1966), still a prominent researcher in the field. A similar review of the Business Periodicals Index showed no activity until 1981. A comparison of the number of AI citations in these two periodical indexes, illustrated in Figure 1, gives rise to three interesting facts.* First, as already implied, the appearance of articles in business periodicals lagged that for applied science and technology by 15 years. Second, of the total number of business oriented articles cited (128), very nearly half (63) were published during the year 1985. And third, since AI first appeared in the business media, the publication growth rate appears to be substantially higher than that for the applied science and technology areas, indicating an impressive buildup of momentum.

Waterman (1985) provides a catalog of expert systems in which he describes the stage of development that each of a total of 179 systems had attained at the time of publication of his book. Although the listing he provides cannot possibly include all existing systems, a compilation of those cited, categorized by application area and stage of development, provides perhaps a relative, if not absolute, indication of the breadth and depth of expert system tools more or less available. The entries from Waterman's catalog (pp 244-299) were tabulated to create Figure 2. The data have been ordered by total number of citations per application area. These same data have been cumulated across application areas and plotted in Figure 3 as a function of development stage only. The area noted as "other categories" includes

---

* This comparison was justified by the basis that both of these index sources are produced by the same publisher, are mutually exclusive in their coverage of periodicals, have similar periodical identification methods in that subscribers (ostensibly libraries and larger corporations) vote as to the source documents to be included in the data base, and represent approximately the same annual total number of citations.

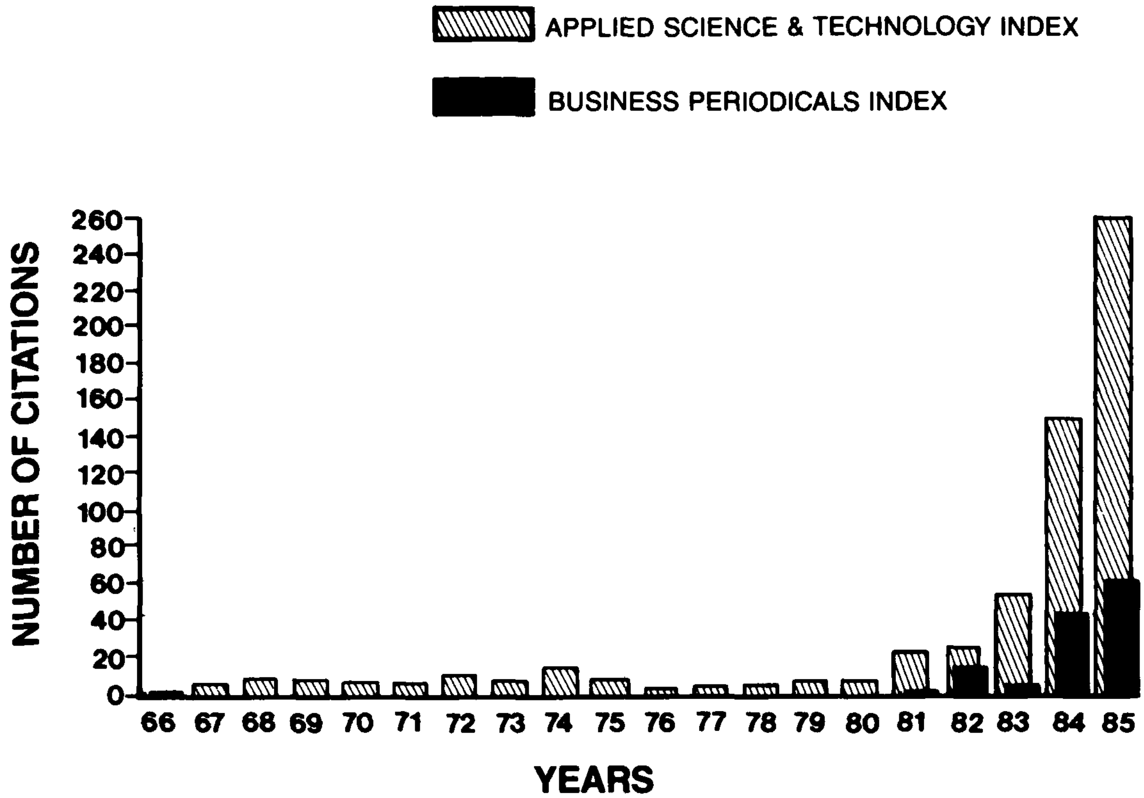


Figure 1. A comparison of number of citations for Artificial Intelligence in Applied Science and Technology Index vs. Business Periodicals Index.

principally systems that were viewed by Waterman as more for exploring reasoning processes, learning by discovery, or as aids in developing expert systems within particular knowledge domains.

For clarity, the application categories in Figure 2 are, reading from left to right: medicine, military science, electronics, chemistry, computer systems, law, information management, engineering, geology, space technology, agriculture, manufacturing, mathematics, physics, process control, and meteorology. It is obvious from Figure 2 that the medical field has been the most heavily invested in, probably because of

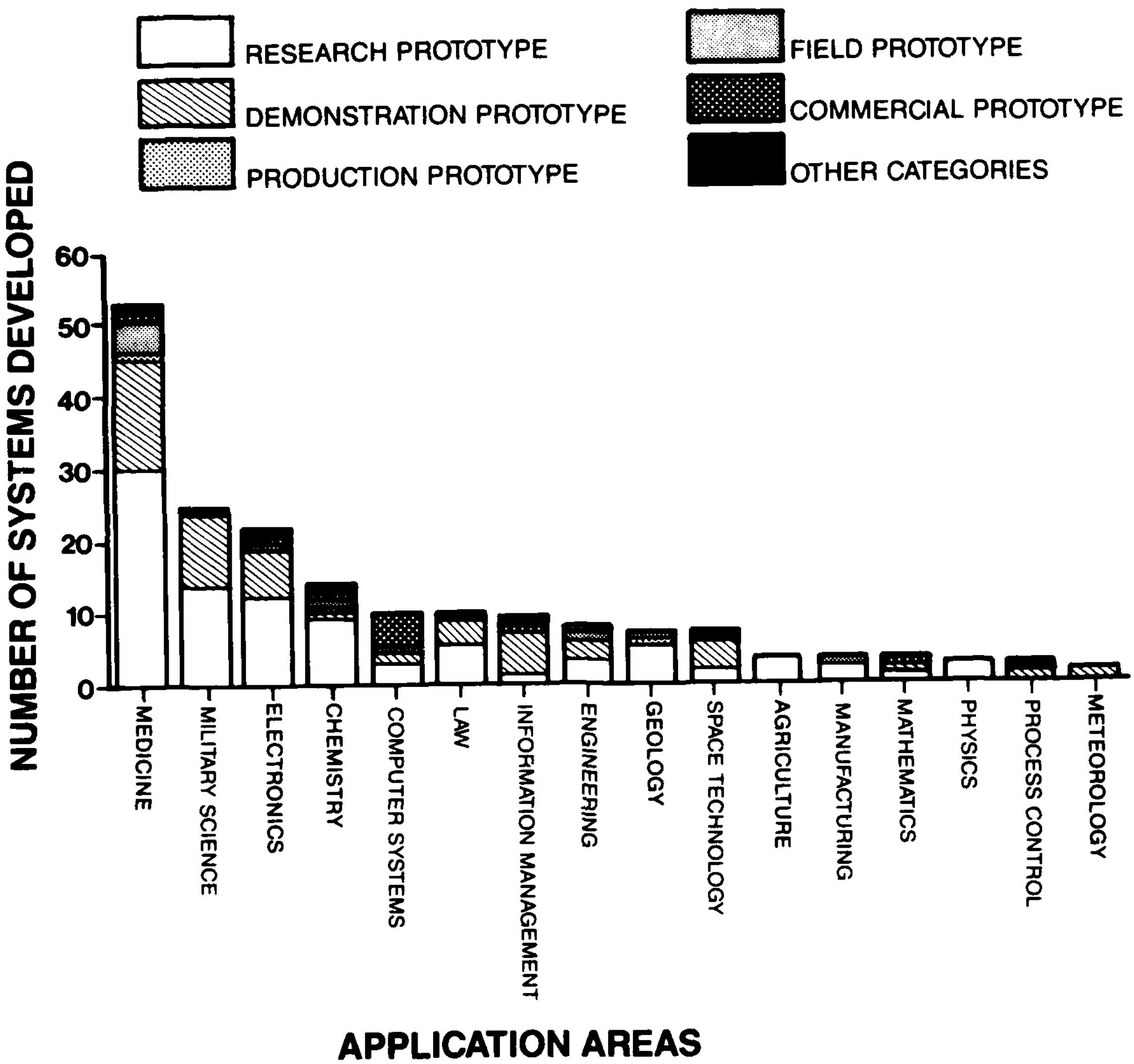


Figure 2. Expert Systems Applications and Stages of Development

the natural diagnostic orientation of typical expert system building tools. Inspection of Figure 2 also reveals that the greatest number of commercially available systems, however, relates to computer system applications. This is not surprising since the people developing the expert systems have typically been computer scientists by training and could be expected to exploit their own technical domain first.

Figure 3 specifically illustrates the preponderance of expert systems that are only at the research prototype or demonstration prototype stage, and the relatively small proportion (9 out of the 179 systems surveyed, or 5.0 percent) of systems that were actually commercially available.

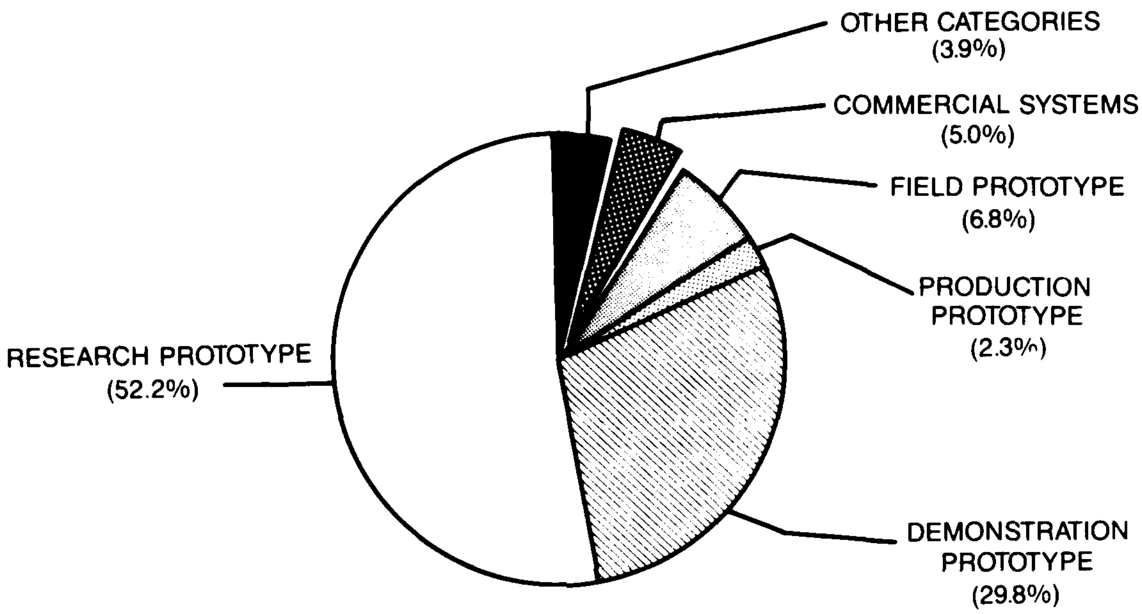


Figure 3. Proportions of Expert Systems at Various Stages of Development

Some Application Examples

To provide the reader with a better idea of what the expert systems that have been developed actually do, how they do it, and how they have been validated, a discussion of the synopsis of expert systems research provided by Duda and Shortliffe (1983) provides an excellent starting point. Their historical perspective over the last 25 years describes the evolution of emphasis in AI as shifting from the early attempts to use computers to exhibit intelligent behavior (conceivably through the identification of few powerful techniques) to a knowledge-based approach which has evolved into the area of expert systems. Classification routines now represent the most successful of expert system programs. In the medical diagnosis area, these are designed to weigh symptom data and provide the clinician an assessment of the most probable causes. Several different approaches for combining probabilistic data have been used, including Bayesian techniques.*

Primary examples cited by Duda and Shortliffe fall into the application areas of medicine, chemistry, computer science, and geology.

MYCIN (used for diagnosis and treatment of infectious diseases) uses a rule-based approach, and as such, must obtain specific information about the patient to formulate both a diagnosis and suggestions for therapy. A rule in MYCIN consists of a series of if/and/then logical statements and the strategy in rule selection is goal-oriented. The program "reasons backwards" from its initial goal (i.e., determination of the cause of infection). At any point in the process, the user may ask the program "why" a particular question was being asked. MYCIN provides both the reason for the question and the rule being pursued, thus allowing the user to trace the reasoning applied. The inability to determine the source of the automated inquiry in some other statistically based diagnostic systems has resulted in user acceptance problems, even though performance of these systems might have been excellent.

Another medical expert system, known as INTERNIST, is used to diagnose problems of internal medicine. Ford (1985) states that INTERNIST can diagnose approximately 500 diseases through analysis of some 3000 symptoms. In an assessment of the performance of this system, 43 diagnostic problems (taken from the New England Journal of Medicine) were correctly diagnosed 25 times by the program, as compared to 28

---

* See bibliography for paper by Duda, Hart and Nilsson (1984) for a treatise on the application of Bayesian methods to rule-based inference systems.

times by the physicians treating the patients, and 35 times by the expert physicians authoring the journal papers.

The DENDRAL program is designed to analyze mass spectral patterns to suggest the chemical structure of unknown compounds. It is based on an algorithm developed in 1964 by the Nobel Prize-winning chemist, Joshua Lederberg. The algorithm starts with a set of mass spectroscopic data and identifies all molecular structures that could be predicted from that set of data (see Harmon and King, 1985). In 1965, Lederberg joined with others at Stanford, notably Edward Feigenbaum, to see if the algorithm could be translated into a set of heuristics that perhaps could produce the same results but much more rapidly. The heuristics were developed through intensive discussions with expert chemists. This information extraction process required approximately 15 person-years and produced not only the DENDRAL program, but also the field of knowledge engineering, as it was termed by its creator, Edward Feigenbaum. The DENDRAL program has led to approximately 50 publications in the chemistry literature and has been effectively validated through routine use by chemists.

The expert system PROSPECTOR is used to identify ore deposits and select drilling sites. Validation of PROSPECTOR was performed in 1980 (see Waterman, 1985) by actually drilling at sites in eastern Washington according to its predictions. PROSPECTOR analyzed the geological, geophysical, and geochemical data describing the area and predicted both concentrations and lack of concentrations of ore-grade molybdenum deposits at particular sites. Several drillings by a mining company confirmed PROSPECTOR's predictions.

The XCON (also referred to as R1) program is considered one of the most successful commercial applications of expert systems technology and is in everyday use by Digital Equipment Corporation, being applied to each VAX configuration sold. Experience across more than 3000 orders processed in one 3-month period showed that over 85 percent of the configurations were flawless, while the remaining were usable after simple corrections. Manuel (1985) states that XCON saves the company approximately $18 million each year, and also results in much greater customer satisfaction.

A major differentiation in the logical approach used by various expert systems has to do with whether the rules in a rule-based system are applied to the facts (or data base), or whether the facts are applied to the rules. The most popular approach (as used in MYCIN, for example) uses a backward chaining strategy in which the data are used to select a number of the most plausible hypotheses (or solutions) in a high-level screening process. The expert system then proceeds with a line of questioning appropriate to acceptance or rejection of each of the hypotheses considered and provides a ranking of possible solutions in their order of likelihood. In contrast, a forward chaining strategy would attempt to reason forward from a given set of facts to identify a solution. The behavior of a forward chaining system can appear to be erratic as it first works toward one solution and then another in its logical sequence. The relatively incoherent series of questions produced by this forward chaining process undoubtedly produces lower user confidence than the "goal-directed" backward chaining approach might. In addition, user queries under a backward chaining system are much more easily satisfied since both the data and the final solution being pursued can be identified to the user, thus providing the rationale for the line of reasoning being used. In order to gain reasonable pilot acceptance, cockpit application of expert systems will almost certainly have to provide explanations to the pilot when requested. Since the backward chaining approach provides this capability most directly, it will most likely be preferred by system designers and pilots as well.

Projection of Expert Systems Technology into the Cockpit Management Arena

Using efforts such as those described above as an experience base for the development of expert systems for cockpit management applications, it seems clear that future efforts must evolve around the identification and encoding of knowledge, requiring the collaboration of pilots, engineers, skilled domain specialists, and behavioral scientists*, together with AI specialists. Strategies for dealing with inference and uncertainty are also reasonable areas for future concentration, as are techniques for adjusting the level of explanations to accommodate a range of user needs. This implies that the program must maintain a model of the user in terms of what he/she knows and what is trying to be accomplished. This fact represents the major difference between expert systems and their closest relatives in the management decision area, so-called decision support systems.

Michaelson and Michie (1983) discuss the differences between expert systems and (procedurally based) decision support systems (DSS). Routine decisions can be modelled by DSS's when they are understood well enough to be able to be specified through mathematical formulae and be procedurally programmed using languages such as BASIC, COBOL, FORTRAN or Pascal. On the other hand, expert systems more typically employ unique symbol processing techniques (using list or logic oriented languages such as Lisp or Prolog, respectively) to incorporate a knowledge base (if/and/then rules) which is processed according to a strict, goal-oriented, deductive inference process in order to achieve a pattern match with the present situation. Ford (1985) characterizes DSS's as helping decision-makers use data and models to solve relatively unstructured problems. As such, the DSS might be in the form of a management information system, or other such data base to which quantitative analysis techniques can be applied by the user. In contrast, the expert system provides a problem-solving capability by codifying the knowledge of experts so that their reasoning, skill, and intuition may be applied to the problem at hand, thereby (hopefully)

---

* The processes of extracting knowledge from experts and modeling the communication needs of the user draw on the skills of the behavioral scientist.

providing a better conclusion or decision. Ford (1985) points out, however, that the DSS provides inherently greater personal flexibility because the user controls all queries, while the expert system may offer little or no flexibility. In any event, the greatest relative advantage of the expert system is its ability to explain its reasoning process by displaying the rules used to make a decision.

Blanning (1984) draws a distinction between decision support systems and expert systems on the basis that DSS's use causal models, as opposed to the judgmental models used by expert systems. Such causal models may treat the areas of production, distribution, marketing or the financial structure of an organization. Blanning suggests that designers of expert systems for management must realize the existence of DSS models and provide interfaces between these and expert system models where appropriate. In discussing existing and possible expert systems for management, he cites several examples that generally fall into three categories of management concern and function that overlap completely with major pilot management tasks in the cockpit: resource allocation, problem diagnosis, and scheduling and assignment. Blanning projects that since resource allocation is such an important managerial function, expert systems will probably be developed to assist in that process (e.g., allocation of an R&D budget to proposed projects, or the preparation of budgets by governmental agencies, as well as by independent firms). In the problem diagnosis area, Blanning envisions the codification of observational data to trigger identification of an impending or real problem. Such codification would lead to rules such as: "If net selling price is more than $2 below budget, sales volume is more than 10,000 units above budget, and profit contribution is more than 5% below budget, then there is evidence (0.7) that price discounts are excessive." Totally analogous rules could be generated for assisting the pilot in identifying otherwise insidious problems with aircraft systems, external threat composition or any other combination of elements that is predictive of a change in situation that the pilot should be made aware of. Blanning (1984) concludes that "developers of management information systems and decision support systems will certainly find in expert system technology a fertile field for research and practice, and developers of expert systems will certainly find management applications a fertile field for research and practice."

Elam and Henderson (1983) also discuss concepts of knowledge engineering applicable to the development of decision support systems. They argue for the incorporation of knowledge engineering techniques to improve the acceptability and utility of DSS's.

A report by Myers (1984) offers the most conservative view of the potential application of expert systems to management operations that could be identified through this review. Opinions voiced by representatives in the banking, computer chip, and computer mainframe industries generally indicated doubts that their "gut-feelings," expertise, and experience in their business worlds could be put into an expert system, since these decision drivers are poorly understood by the experts themselves.

It should be noted, however, that perhaps the management representatives would have been more enthusiastic about expert system applications to their speciality areas if they had had some experience with the process of information extraction from their respective areas through the help of someone specially trained to assist in that process. Such (knowledge engineers) have obviously been the cornerstone of knowledge base developments within the most successful expert systems.

One may hypothesize many scenarios for the application of AI and expert systems to cockpit management problems. The most reasonable forecast of future investments in AI for cockpit management applications would seem to be those that play a role in the training process, have relatively short-term payoff, and are perceived to be cost effective. Based on these criteria, the most likely candidates for earliest implementation would seem to be systems for speeding up and/or simplifying the tasks of mission planning (and replanning, when an unexpected event occurs during the mission), navigation and navigation systems management, diagnostic systems, and threat systems management. Indeed, it will be shown in Chapters 2 and 3 that these and many other application efforts are in various stages of development.

From an industrial management perspective, Harmon and King (1985) portend that, in the coming years, problem solving and decision making will be automated just as surely as production lines are, and that expert systems will be the "robots" of middle management. They project that tasks thought to be impossible to computerize will become amenable to computer solutions and jobs and individuals associated with older ways of problem solving and decision making will change or be replaced. They conclude: "As with past technologies, those individuals that can blend the power of these new technologies with the necessities and the constraints of their organizations will be the winners."

Conclusions

It seems that the major challenge to aircraft system designers will be to determine how the advanced technology of AI and expert systems can best be woven into the fabric of their technology areas so that improvements in man-machine and operational effectiveness are demonstrable. The key to success of such ventures will most likely be the extent to which pilots find the systems rewarding to use in terms of reducing workload during critical mission phases, increasing their situational awareness, and improving their chances of survival under dangerous flight regimes or threat environments.

In the process of developing this report, the author was continually reminded of the words of Allen (1978) regarding the importance of information processing to the advancement of science. He states: "Information processing is the essence of scientific activity. As physical systems consume and transform energy, so too does the system of science consume, transform, produce, and exchange information." What more satisfying approach to information exchange could there be than one in which the expert system becomes the vehicle for transfer of knowledge and technology from the senior pilot, engineer, or scientist to his less experienced colleague in the cockpit of tomorrow's aircraft.

Bibliography

Allen, T.J. (1978). Managing the flow of technology: Technology transfer and the dissemination of technological information within the R&D organization. The MIT Press, Cambridge, MA.

Applied Science and Technology Index, (cumulative quarterly and annual issues). H.W. Wilson Company, 950 University Avenue, Bronx, NY 10452.

Blanning, R.W. (1984). Management applications of expert systems. Information and Management, 7, 311-316, Dec.

Business Periodicals Index (cumulative quarterly and annual issues). H.W. Wilson Company, 950 University Avenue, Bronx, NY 10452.

Duda, R.O., Hart, P.E., and Nilsson, N.J. (1984). Subjective Bayesian methods for rule-based inference systems. In O. Firschein (ED.) Artificial Intelligence, Vol VI (pp 97-104). AFIPS Press, Reston, VA.

Duda, R.O. and Shortliffe, E.H. (1983). Expert systems research. Science, 220, 261-268, 15 Apr.

Elam, J.J. and Henderson, J.C. (1983). Knowledge engineering concepts for decision support system design and implementation. Information and Management, 6, 1136-1137, 14 Sep.

Engineering Index (published monthly). Port City Press, Baltimore, MD, 21208.

Ford, N.F. (1985). Decision support systems and expert systems: A comparison. Information and Management, 8, 21-26.

Harmon, P. and King, D. (1985). Artificial intelligence in business. John Wiley & Sons, Inc., New York, NY.

Manuel, T. (1985). The pell-mell rush into expert systems forces integration issue. Electronics, 58, 54-59, 1 Jul.

Michaelson, R. and Michie, D. (1983). Expert systems in business. Datamation, 29, 240-244, Nov.

Minsky, M.L. (1966). Artificial intelligence. Scientific American, 215, 246-252, Sep.

Myers, E. (1984). Business takes the fifth. Datamation, 30, 53+, Sep.

Waterman, D.A. (1985). A guide to expert systems. Addison-Wesley, Reading, MA.

CHAPTER 2

Artificial Intelligence Development Applications in DARPA, DOD and NASA

Introduction

This section reviews ongoing and planned efforts to develop a broad range of AI (including expert systems and robotics) applications by various government agencies. Funding of these efforts has been spurred by the burgeoning computer technology, together with concerns by the various organizations that today's data processing, analysis, and decision making requirements may be performed more reliably and efficiently by programs specially designed to use the codified knowledge base of experts. Harsh or unfriendly environments provide yet another setting where AI developments in the field of robotics find great user support.

It will be demonstrated that we are at *the threshold of a surprising* range of application capabilities which may eventually dramatically reduce the present required reliance on real-time human judgement. Exploration of AI technology and the development of expert systems offer the opportunity for considered, higher quality judgements, but without the time lags usually associated with ponderous rumination, especially by those at the lower decision making levels.

It is important to recognize at the outset that these systems do not think or reason in the way that humans do. When design and technology permit, however, they may allow relatively complex information processing and decision processes to be performed rapidly and accurately under situations that man may find difficult or impossible.

DARPA Sponsored Efforts

The Defense Advanced Research Projects Agency (DARPA) has embarked on an ambitious program to develop computer technology that will perform at several orders of magnitude beyond present systems. Ulsamer (1985) quotes Dr. Robert S. Cooper, past DARPA director, as estimating processing power of these machines to be up to 10,000 times as great as the largest current-generation computers. This development effort is part of DARPA's Strategic Computing Program which will spend approximately $600 million over its first five years. Klass (1985) describes DARPA's overall program goals as aiming to "provide the U.S. with a broad line of machine intelligence technology and to demonstrate applications of the technology to critical problems of defense." Three application programs have been initiated to assist in the technology demonstration process. The first of these is the autonomous land vehicle (ALV) program which is designed to provide a "strong pull for vision and image understanding technology." The ALV is to use a single TV camera, and through processing routines, determine the path of a road and follow it automatically. Although initial demonstrations will involve only about 20 feet of travel before the vehicle must stop and recompute the road, successive planned increases in computational power over the following year will allow a continuous speed of about 10 kilometers per hour, with 60 kph as the target speed by the end of the program. It is also planned that this eight-wheeled, 5000 pound vehicle will eventually have a five-color laser scanner/radar that will allow it to detect obstacles by measuring their absorption and reflection at the five different wavelengths. This will allow it to traverse open terrain. The on-board expert system will plan routes by combining a digital terrain map with sensed environmental data, plot strategies to avoid obstacles, update its own terrain data base and generate all steering and speed commands. DARPA estimates that such a system will require approximately 6,500 rules, together with an average execution rate of 7,000 rules per second. A single execution consists of a complete cycle of examination, interpretation, and response to one rule in a particular situation.

To support the four orders of magnitude increase in computational power required to process all the required data, DARPA is also investing heavily in gallium arsenide (GsAs) and gallium aluminum arsenide (GaAlAs) material production technology. These materials are superior to the conventional silicon materials used in today's computer chips in terms of their tolerance to radiation from nuclear weapons, as well as their much wider operating temperature range. Rockwell International and Honeywell are the contractors involved in this technology development.

Wallace et al. (1985) describe the first results obtained in continuous motion road-following tests for the AVL, performed by the Robotics Institute of Carnegie-Mellon University. Preliminary vision and control systems have been developed which allow traversing of an outdoor path at 2 cm/sec with an image processing time of 2 sec/image. Although locomotion rate is low, it should be realized that this represents the first <u>complete</u> system, including the low-level drive motors and the top-level control loop and user interface.

The second major thrust in DARPA's Strategic Computing Program is the Pilot's Associate Program. Stein (1985) states that the Pilot's Associate will provide expert advice regarding situation assessment, mission planning, systems status, and tactics to the pilot. A key aspect of this program will be a natural language interface with the pilot. There are four initial expert systems to be developed. The situation assessment manager will assess the tactical environment (i.e., weather, terrain, targets, threats, and the capabilities of on-board systems to meet the demands of the mission). The tactical planning manager would recommend a best course of action, given the present tactical situation. The mission planning manager would assist the pilot in maintaining, managing, and revising mission plans to accommodate the present situation. The systems status manager would identify malfunctioning, non-operational, or impending system problems and attempt to transfer the lost or nearly lost function to another system or systems. The Pilot's Associate Program is being directed by the Air Force Wright Aeronautical Laboratories (AFWAL) at Wright-Patterson Air Force Base, Ohio.

The last component of DARPA's Strategic Computing Program is a battle management system for both land-based and shipboard use on a Navy carrier task force. A facility will be developed for capability demonstrations in Hawaii, next to the CincPacFlt command center. The objective of this program is to develop expert systems to allow a command center to perform 96-hour look-ahead contingency planning, and communicate with the system using natural language. Texas Instruments is the integration contractor for the Hawaii facility, and Bolt Beranek and Newman is developing the natural language subsystem. Carnegie-Mellon University is developing a situation assessment expert system, and Computer Corp. of America is developing a graphic interface display which will be installed shipboard.

Computer hardware and software to support these various efforts are being developed through several joint efforts by universities, industrial research laboratories, and entrepreneurial companies with the hope that greater numbers of students will become interested in the advanced computer processing techniques this program will require and thus build our national resources and capabilities in this area.

The Mitre Corporation, under funding by the Rome Air Development Center, has developed a laboratory-based expert system to analyze tactical intelligence data. The system, known as "Analyst" (see Freck and Bonasso, 1985) is programmed in Lisp and incorporates rules generated through interviews with tactical intelligence experts. To transform the laboratory-based system into a useful operational tool, Mitre feels the system needs to be put into the field and adapted to the specific needs of a field command, as well as be made more user-friendly to accommodate the average GI intelligence analyst. Since Analyst employs a user developed data base created via keyboards and graphics, information and rules can be modified to adapt the system to a new tactical need. DARPA has been sufficiently impressed with the progress thus far to team with Mitre to perform two field experiments with the system to examine user interaction issues. DARPA will apply the results to its Space Strategic Computing Program.

Buffalano (1985) states that DARPA will have spent $15-20 million in expert systems efforts in FY85. It is anticipated that over the next three years DARPA will spend $4 million to develop the "New Generation Expert System" which will be a modular and generic system, supposedly as easy to use as a "user-friendly" spreadsheet or data base management system. The new-generation system will incorporate a natural language interface and sophisticated graphics as input-output media.

NASA Sponsored Efforts

Wolfe (1985) describes several roles for AI and expert systems supporting future shuttle and space station functions. A total of eight NASA divisions are working to develop systems to be used for payload processing, fluid system management (e.g., loading liquid oxygen), tracking, data acquisition, electrical power system management, network control, robotics and simulation, to name a few. NASA is projecting a technology freeze date* of 1987 for application of these systems to the space station, even though the first launch is not scheduled until 1992.

Expert systems development efforts at Johnson Space Center are further described by Marsh (1985). Among those discussed for space station or shuttle applications is the Flight Design System (FDS) in which the user will tell the program what the mission goals are, and FDS will calculate the appropriate launch information, such as when and what

* The technology freeze date refers to the date at which the design of systems can no longer be changed.

direction to launch, as well as what altitude to perform particular functions. The NAVEX (Navigation Expert System) will perform the work of three controllers monitoring and routing the data from radar stations tracking the space shuttle. In an evaluation of NAVEX during mission 41-C, the program correctly rerouted weak signals from the shuttle when television news transmitters interrupted data from the shuttle as it landed. NAVEX performed the rerouting faster than the controllers. Another system, known as FIXER (Fault Isolation Expert for Enhanced Reliability), is designed to identify faults in the shuttle's system for removing carbon dioxide from cabin air. The RENEX (Rendezvous Expert System) will perform on-board mission planning, together with fault isolation and recovery functions under nonideal conditions. NAVEX and RENEX were created using the expert system shell called Automated Reasoning Tool (ART) from Inference Corporation, Los Angeles, CA and run on a Symbolics 3600 computer. FIXER was developed using another shell called Knowledge Engineering Environment (KEE), developed by Intellicorp of Menlo Park, CA.

A prototype AI deep-space mission planner has been developed by the Jet Propulsion Laboratory for unmanned spacecraft such as Voyager and is described by Vere (1985). The system runs on a Symbolics 3600 computer, works 10-50 times faster than a human analyst, and is called "Devisor." Spacecraft sequences for planetary encounters are generated by the system on the basis of goals input to it. For example, a goal might be to obtain a recording with an instrument pointed in a particular direction, within a particular time window. Devisor would then consult its knowledge base to retrieve the set of actions, events, and inferences needed to accomplish the goal. An action might be to move the instrument platform to a specific orientation and position the tape recorder to a predetermined point. An event could be the stoppage of camera platform oscillations after movement, and an inference might be the interpretation of that state. Devisor generates a plan of action for each new set of goals. This plan resembles a PERT* chart with critical path and float times identified. The intent with this system is to transmit only the goals to the spacecraft and depend on Devisor to generate an acceptable plan. This design eliminates the transport delays that would normally be encountered if signals had to be transmitted back and forth between the ground and the spacecraft. Devisor creates a plan by working backwards from its knowledge base to find an action that matches each of the goals. This is known as a "best-first search process" and is the basis for generation of an acceptable, as opposed to an optimal, plan.

Grenander (1985) provides additional insight into the development of Devisor. First, transferring the program from a DEC PDP-10 computer to a Symbolics 3600 computer with 4-million bytes of memory reduced nominal processing time from 2 hours to 8 minutes. Secondly, as has been observed in most other expert system development efforts, knowledge engineers had to work closely with the domain experts. In this case, information from a total of seven technical and management specialists had to be integrated, including that from two AI researchers, a spacecraft specialist, a sequencing expert, and three managers at various project levels. Thirdly, development of such a system is not cheap. Devisor, together with two smaller systems (one to display where the spacecraft is with respect to satellites and planets, and the other to display a color-coded timeline to allow the user to decide when external constraints would permit an observation) required approximately four years and $3 million to bring them to the prototype demonstration stage. Due to the success of the project, however, future developments are planned which will provide greater system autonomy for a Voyager type vehicle. These include the capabilities to monitor both progress toward a planned goal, and fault diagnosis to assess the cause of goal-threatening situations.

---

* PERT stands for Program Evaluation and Review Technique and is commonly used in the management of system development efforts to identify where critical dependencies exist among event sequences.

According to the April 22, 1985 issue of Aviation Week and Space Technology (no author specified - pg 83), the NASA Advanced Technology Advisory Committee recommended accelerated development of AI software and robotics for space station applications. Specifically, the group wanted 13% (or $1 billion) of the total space station cost to be spent on expert systems, robotics and other forms of automation.

It is important to point out that the AI and expert systems that are being developed and demonstrated by NASA are not programs that try to learn. Instead, they are programs that imitate the plans and decisions of experts that have already been proven (see Marsh, 1984)*.

Air Force Sponsored Efforts

The April 22, 1985 issue of Aviation Week and Space Technology also provided an overview of advanced technology initiatives at the Air Force Wright Aeronautical Laboratories (AFWAL) at Wright-Patterson AFB, OH. Preliminary workshops at the Avionics Laboratory defined the AI application areas of: a) vision and image understanding; b) generic aerospace electronics systems; c) design automation; and, d) maintenance and diagnostics, as primary candidates to be evaluated. It is intended that the first efforts be performed through basic and applied research contracts with universities and industry.

Anderson et al. (1984) describes a prototype expert system that would be incorporated into a Pilot's Associate to handle in-flight emergency procedures in an advanced tactical aircraft. The total system includes the expert system itself (which contains the knowledge base, inference engine, and message display) together with an environmental simulation that contains the controls, displays, and other interface

---

* The essence of this comment came from an interview by Marsh with Robert H. Brown, Chief of Technology Development for NASA.

elements required between the expert system, aircraft subsystems and the pilot. An example of how the system would function in response to an in-flight emergency in the form of an F-16 canopy loss is described. In this situation, reducing airspeed to 180 knots, lowering the pilot's seat, setting breathing oxygen to 100%, and, depending on the criticality of fuel, extending the flaps, may all be performed by the expert system.

An expert tactical navigation system (the "Expert Navigator") is described by Pisano and Jones (1984) in which a knowledge-based approach to the management of navigation sensors is being evaluated using a high-fidelity F-16 simulation of a deep interdiction mission. The effort is being performed as part of the Adaptive Tactical Navigation Program by the Analytic Sciences Corporation, Reading, MA, under contract to the Air Force Avionics Laboratory. The objective of the program is to improve the utility and management of the navigation sensor suite, including GPS (Global Positioning System), INS (Inertial Navigation System) and digital terrain aids (e.g. SITAN - Sandia Inertial Terrain - Aided Navigation). Expert Navigator consists of two parts, a "Resource Manager" and a "Mission Planner." The function of the Resource Manager is to monitor the performance of the navigation sensor suite (degradation may emanate from internal or external causes) and relate present operating characteristics to planned mission events. For example, an elevated ECM (Electronic Countermeasure) level may be sensed and cause a reduction in the judged validity of position information. Similarly, missed turnpoints or overflight position discrepancies may be identified by the Resource Manager. When such events occur, they trigger a "concern indicator" (signalling that something unexpected/ unplanned has happened that may threaten successful accomplishment of the mission) which then fires a series of events designed to determine the probable cause of the problem and what to do about it. Both the concern indicators and the proper courses of action (rules) were developed after extensive interviews with Air Force aircrewmembers. Each of eight concern indicators selected for demonstration can trigger

30 to 50 rules. The second half of the system, the System Planner, continuously projects the capability of the navigation suite (at its present capability level) to support the primary mission and suggests alternatives when the primary mission cannot be supported. The System Planner uses the projected accuracy of the navigation system, together with its prestored data regarding the target, and waypoints to it, on-board weapons and their delivery modes, threats, and planned altitudes for the various legs of the mission, to compute a projected CEP (circular error probability) for each available attack option. By comparing the computed CEP with that required to successfully eliminate the targets, a prioritized set of attack options is generated. Pilot interface with the system is through a voice recognition system. Results of this effort will feed into DARPA's Pilot's Associate Program.

Lineback (1985) describes an interactive flight simulation work station (called Imaps - for Interactive Mission Analysis Planning Station) being developed by Merit Technology Inc., Dallas, TX. The purpose of this work station is to combine commercially available graphics software for out-the-window scene generation with digital terrain data from the U.S. Defense Mapping Agency to produce a visual flight simulation capability at far lower cost ($50,000 to $500,000, depending on simulation complexity) than conventional mainframe-based flight simulators. The author projects that Imaps could be used to train pilots on expert systems and other AI based avionics. Pilots would be able to practice flying the mission with known threat placements in the data base prior to the actual mission. The Imaps system is being integrated for use by several defense contractors and government agencies.

The last, and most futuristic Air Force effort to be discussed here, concerns battle management, which is recognized in the Strategic Defense Initiative (SDI) Program as one of the most challenging technical aspects facing its developers (see Ropelewski, 1985). This awareness resulted from SDI architecture and candidate technology studies conducted independently by 10 industry teams, each funded at $1 million. A second phase effort will be performed by four or five of these teams to further narrow the field of technologies and architectures.

Navy Applications

Aside from the DARPA supported battle management system for land-based and shipboard use described in the DARPA applications section of this report, no U.S. Navy AI developments were identified in this review. However, an expert system for evaluating electronic warfare (EW) tasking plans for the Royal Navy is described by Gadsden (1984). This system is designed to provide advice on allocation of EW equipment and uses a backward-chaining Bayesian-inferencing expert system shell to develop inference nets similar to those used by the Prospector system (see Waterman, 1985) to identify ore deposits and drilling sites. The tasking involves the proper/reasonable assignment of frequency bands for jamming enemy radar equipment, once radar emissions are detected. The conventional (manual) approach requires the planner to determine what enemy ships and aircraft might be in an area, what equipment they might be carrying, what their emission spectra are, and what allocation of jamming equipment and frequencies might best defend his own force. This manual procedure involved looking up a great many numbers in several printed volumes, a time consuming, tedious, error prone, and judgmental process. The EW tasking advice system consists of the expert system, a data base containing equipment characteristics and tasking rules, together with a color display and keyboard for operator interaction. With it, the operator places symbols on the screen corresponding to the enemy ships and aircraft he expects, together with similar data for his own composite of ships and aircraft. The equipment data base is accessed automatically and allocations are made according to the rules provided by the knowledge base. In the process of developing this system, special attention was paid to the man-machine interface design, facilitating use of the data base, interactive display graphics, and algorithm portrayal.

Army Sponsored Efforts

Campen and Gorden (1984) discuss the application of AI to Army tactical operations. They posit that the biggest threat we face in a next world war "is that NATO may lose the first key battles because it has too much data." They argue that the ability to acquire needed information and use it effectively will provide a stronger advantage than numerical superiority of combat forces. The designs of present day intelligence data processing systems suffer from having been designed for other intelligence users, rather than for commanders and operations officers. Heuristic expert systems are needed to supply the commanders and operations people the information they need. By means of a fictitious tactical scenario involving the use of an expert system by a Soviet commander, Campen and Gorden illustrate the deductive power such a system could have when meshed with intelligence reports from that commander's ground and airborne forces throughout the course of the battle. Rules and relationships used by the expert system were presumed to have been compiled from U.S. and NATO doctrinal literature, professional literature, satellite photography, and from monitoring our radio broadcasts and observing NATO exercises and training. The picture these authors paint strongly supports DOD development of such systems to increase the productivity of the best analysts, and helps inexperienced or poor analysts perform as well as the best analysts performed previously.

The Army is especially interested in developing robots for dangerous battlefield tasks, such as clearing mine fields and handling heavy ammunition. The May 1985 issue of Data Processing (pg 47) describes the Battlefield Robotic Ammunition Service System (BRASS)*. This system

---

* The original source of the information presented in this article is from the report entitled "Artificial Intelligence and Robotics in Military and Paramilitary Markets", IRD, 6 Prowitt Street, Norwalk, CT 06855.

locates, carries, and positions a pallet of ammunition next to a gun site. It then installs fuses into the rounds and prepares them for firing. The system is expected to be especially beneficial during very hot or very cold weather when soldiers performing these tasks become highly fatigued.

Summary

As a final note, the May 1985 issue of Data Processing states that more than 40 companies in the U.S. are developing various types of AI-based military and paramilitary robots, and that annual production of these systems is projected to pass the half billion dollar mark by 1994. By then, an additional $1 billion per year is also expected to be spent on research and development of military robots and AI.

Bibliography

AI in military development (1985). Data Processing, 27 47, May.

Air Force Systems Command accelerates R&D efforts (1985). Aviation Week and Space Technology, 67, Apr 22.

Anderson, B.M., Cramer, N.L., Lineberry, M., Lystad, G.S., and Stern, R.C. (1984). Intelligent automation of emergency procedures in advanced fighter aircraft. In Proceedings of First Conference on Artificial Intelligence, (pp 496-501), IEEE Computer Society Press, Silver Spring, MD 20910.

Buffalano, C.A. (1985). Expert systems for the military. Aerospace America, 41-43, Apr.

Campen, T. and Gordon, D.E. (1984). Application of artificial intelligence to tactical operations. In Proceedings of the Army Conference on Application of Artificial Intelligence to Battlefield Information, White Oak MD, Apr 20-22, 1983, (pp 81-80). DTIC No. AD P003025. (Note: entire Proceedings identified as DTIC No. AD-A139685.)

Freck, P.G. and Bonasso, R.P, (1985). Drawing a clear picture of the battlefield. Aerospace America, 46-57, Apr.

Gadsden, J.A. (1984). An expert system for evaluating electronic warfare tasking plans for the Royal Navy. In Proceedings of First Conference on Artificial Intelligence, (pp 86-91), IEEE Computer Society Press, Silver Spring, MD 20910.

Grenander, S. (1985). Toward the fully capable AI space mission planner. Aerospace America, 44-46, Aug.

Klass, P.J. (1985). DARPA envisions new generation of machine intelligence technology. Aviation Week and Space Technology, 46-54, Apr 22.

Lineback, R.J. (1985). A cheaper way to learn if avionics gear will fly. Electronics, 58, 28-29, Dec 16.

Marsh, A.K. (1984). Pace of artificial intelligence shows acceleration. Aviation Week and Space Technology, 24-25, Dec 10.

Marsh, A.K. (1985). NASA space station effort drives expert systems research at Johnson. Aviation Week and Space Technology, 59-61, Apr 22.

NASA report urges development robotics, software for station. (1985). Aviation Week and Space Technology, 63, Apr 22.

Pisano, A.D. and Jones, H.L. (1984). An expert systems approach to adaptive tactical navigation. In Proceedings of First Conference on Artificial Intelligence, (pp 460-464), IEEE Computer Society Press, Silver Spring, MD 20910.

Ropelewski, R.R. (1985). Battle management, C^3I network challenge resources of SDI office. Aviation Week and Space Technology, 19-21, Jul 15.

Stein, K.J. (1985). DARPA stressing development of pilot's associate program. Aviation Week and Space Technology, 69-74, Apr 22.

Ulsamer, E. (1985). The next computer generation. Air Force Magazine, 87-93, Jun.

Vere, S.A. (1985). Deviser: An AI planner for spacecraft applications. Aerospace America, 50-53, Apr.

Wallace, R., Stentz, A., Thorpe, C., Moravec, H., Whittaker, W., and Konade, T. (1985). First results of robot road-following. In Proceedings on Artificial Intelligence, Vol 2, (pp 1089-1095), Kaufman Publishers, Inc., Los Altos, CA 94022.

Waterman, D.A. (1985). A guide to expert systems. Addison-Wesley, Reading, MA.

Wolfe, A. (1985). How NASA will use AI in space. Electronics, 58, 32-37, Sep 16.

Wright Laboratories broadens advanced technology initiatives. (1985). Aviation Week and Space Technology, 77-84, Apr 22.

CHAPTER 3

State-of-the-Art Review and Projection of Future Expert System Developments

Introduction

The remarkable recent expansion of interest in expert systems has been kindled by the burgeoning computer hardware technology, growing interest in easier to use software languages and programs, and motivation by DOD, industry, and academia to perform certain tasks more efficiently. In many cases, basic research from university AI laboratories has spawned impressive advancements in processing capabilities, owing to rather radical departures in computer architecture and algorithm design. A notable example is the Symbolics 3600 family of processors (designed at MIT and marketed by Symbolics Inc., Chatsworth, CA) having the capability to process symbolic information using the Lisp language, as well as Prolog, Ada, Fortran-77, and Pascal. These systems represent today's state-of-the-art and are finding growing applications in expert systems development, as well as industrial automation, computer-aided electronic circuit design (especially VLSI -Very Large Scale Integrated circuits), and automated management of details to help control complexities in development of complex software systems.

The purpose of this section is to provide insights into a broad spectrum of technology developments, some of which relate to the power of computational processing, while others deal with the real, potential, or forecasted application of this technology. Since the field of AI in general, and expert systems in particular, owes its foundation to academic study of basic scientific issues regarding reasoning processes and the reflection of these in prototype programs, continuing development of this area from a basic research, as well as a philosophical perspective is described. The opinions of those more critical in their philosophical and experiential perspectives are also discussed.

Finally, since the amount of information in this section is considerable and represents various positions that at times may seem congruent and at other times at odds with each other, a summary section is provided to help the reader gain a "thumbnail" impression of where we are now, what we know, and what needs to be done.

Industrial Expert System Developments

The February 17, 1986 issue of Aviation Week and Space Technology contained a technical survey of artificial intelligence (pg 40-92) and provided an excellent update of military as well as industrial developments. Significant strides are being made to increase the capacity and reduce the size of symbolic processing hardware. For example, Texas Instruments (TI) reported their development of a 32-bit Lisp language computer chip, measuring 1 cm sq., containing more than 500,000 transistors. This single chip contains the equivalent of about two-thirds of the components in the central processing unit (CPU) of TI's Explorer symbolic processor (introduced in 1984) having two 11 X 14 in. CPU boards in a package measuring roughly 3 cu. ft. The new chip will be delivered to DARPA as part of a demonstration compact Lisp machine having an eight-megabyte capacity. TI's intentions are to use the Lisp chip as the heart of the microprocessor in a variety of symbolic computing applications. The compact Lisp machine will be built in a 3/4-ATR (airborne transportable rack) configuration (7.5 X 7.63 X 12.51 in.) for standard rack mounting. Corporate estimates at TI project there will be 20 to 30 expert systems onboard an F-16 or an ATF (Advanced Tactical Fighter) of the future, each requiring its own dedicated Lisp machine.

Since the architecture of the compact Lisp machine is identical to that of the Explorer, TI used an Explorer system to demonstrate an

Emergency Procedures Expert System (EPES) for F-16 applications involving multiple emergencies. The first demonstration treated a simultaneous loss of canopy at altitude and failure of the engine shaft. Supposed appropriate courses of action, gleaned primarily from flight manuals (over a seven-month period), were incorporated into the knowledge base. When these were shown to an F-16 pilot, he reported that, although TI had accurately extracted the technical information and procedures, "no living pilot would do it that way." Subsequently, another 11 months were spent redeveloping the knowledge base, this time using pilot interviews as data. As a subcontractor to McDonnell Aircraft Co. (McDonnell and Lockheed Georgia were dual winners of DARPA's Phase I contract for the Pilot's Associate Program), TI will draw from its EPES experience to develop the Battle Management Program, which is also being funded by DARPA.

The McDonnell Aircraft Co. (MCAIR) has a number of internally funded programs designed to support aspects of the Pilot's Associate Program as well as other AFWAL interests. These include: a) intelligent avionics management systems, b) a reconfiguring flight control system, c) an airborne threat engagement management system, d) an inertial navigation system fault analysis and management system, e) an intelligent sensor allocation and cueing system for air-to-air attack, f) aircraft maintenance aids, and g) mission route planning. MCAIR feels that putting AI techniques into the cockpit will be the most difficult application to actually achieve, due to the rapid response required to pilot queries, and have therefore initially invested most heavily in maintenance aids and route planning. In addition, they have completed a two-year effort to develop an AI system to advise a pilot on how to reconfigure his sensor suite based on partial failures. Significant progress has also been made on the flight control reconfiguration problem. A set of control laws was developed based on the experiences of a flight control expert. These laws are used to mimic the flight control expert's advice concerning the optimum combination of control parameters that the pilot needs to manipulate in the face of present aircraft capabilities and mission requirements.

The Boeing Co. has established an Artificial Intelligence Center (Bellevue Washington) in which several initiatives are being explored. Among these are efforts to develop: a) expert systems for space station operations, b) "brilliant" munitions that would, for example, use pattern recognition in combination with the expert system to verify targets or autonomously change targets in flight, c) knowledge-based software engineering for generating expert systems, d) robotics for automation of aircraft production, and e) expert systems as pilot decision aids. Boeing is using Intellicorp's KEE (Knowledge Engineering Environment) expert system shell to develop expert systems for various command, control, communications and intelligence (C^3I) applications. One system is being developed for the AWACS (Airborne Warning and Control System) aircraft to assist the radar operator in correlating information from electronics support sensors. Another is being developed to assist the Tactical Officer aboard the Navy's P-3C aircraft in the placement of sonobuoys for submarine detection and tracking. The KEE shell is implemented in Lisp and run on Boeing's Symbolics 3640 and 3670 computers. Some of the other programs (such as the "brilliant" munitions research) are being performed using the programming tool called Automated Reasoning Tool (ART) from Inference Corp. (Los Angeles), which is also implemented in Lisp and run on the Symbolics machines. The market costs of the KEE and ART packages in 1986 were $10,000 and $15,000, respectively.

New high-speed computers, designed especially for expert systems applications, are in the development planning process as part of DARPA's Strategic Computing program. The February 17, 1986 issue of Aviation Week and Space Technology (pg 45-52) quotes Saul Amarel, Director of the Information Processing Techniques Office at DARPA, as saying his office is funding programs to develop "more advanced computers designed for symbolic logic AI and signal processing applications." BBN (Bolt Beranek and Newman), Inc. already has a symbolic logic computer on the market (called Butterfly) that can be configured with up to 256 microprocessors. DARPA is sponsoring the upgrade of this system to double its speed by replacing the original Motorola M-6800 microprocessors with M-68020 (32-bit) units, each having four megabytes of memory. In addition, this enhanced Butterfly will use a Motorola M-68881 processor to allow floating-point calculations, which will help improve speed further.

Similarly, Carnegie-Mellon has developed a very high speed computer (called Warp) designed to handle signal processing as required to extract information from imagery. Warp would be used in conjunction with a symbolic logic computer to achieve specific goals. Warp is referred to as a systolic processor, having multiple array processors, each consisting of multiple microprocessors. DARPA has funded General Electric Corp. to build seven of the Warp machines, each having 10 array processors. Each of these processors will perform 10 million 32-bit arithmetic floating-point calculations per second (i.e., 10 megaflops), for a total system top processing speed of 100 megaflops.

Thinking Machines, Inc. of Cambridge, MA has developed the "Connection Machine" for symbolic processing, which has been demonstrated to operate successfully with 64,000 individual microprocessors, each with four kilobits of memory. The design of the Connection Machine makes it "very amenable to using wafer-scale integrated circuits," a characteristic which, according to DARPA's assistant director of Information Processing Techniques Office, Stephen Squires, "could provide very advanced symbolic logic capabilities in a very small size." Squires projects that by combining new signal processing techniques as represented by systolic arrays and signal processors having programmable interconnections, operating speeds of 10-50 billion floating-point operations per second (gigaflops) should be possible within the next several years.

An AI testbed is being funded jointly by DARPA and the Army which will use a Butterfly processor to study how best to analyze and make decisions on the information content of various types of electronic reconnaissance imagery (e.g., FLIR - forward looking infrared) as well as help deduce, based on the identified targets, what type of military organization is involved and where its headquarters is likely to be found as a function of enemy doctrine and tactics. DARPA has a similar effort underway with Hughes Aircraft Co. to extract target information from reconnaissance photoimagery.

In another demonstration program, DARPA will apply expert system techniques to seismic data from an array of sensors located around the world in order to detect when a low-level nuclear test has occurred. Ensco, Inc. and Teknowledge Inc. will jointly develop the expert system, ostensibly in conjunction with the experts at the Air Force Technical Applications Center, Patrick AFB, Florida, who now perform the function manually, but are not able to cope on a real-time basis with the extraordinarily high data rates (up to 10,000 bits of data per second) of high quality seismic sensors.

Grumman is also exploring several technology development areas, according to the Aviation Week report (pg 83-85). Three major technology thrusts are discussed. The first of these is the ALERT (Algorithm, Learning, Evaluation, and Recognition Technique) system in which optimum mathematical algorithms can be developed for specific applications such as aircraft or speech recognition using a library of pattern recognition algorithms. The system allows significant reductions in classification algorithm development time. Their second thrust is in the area of expert maintenance diagnostic systems. In a study with BBN, Inc., Grumman found AI techniques significantly reduced false alarms associated with built-in-test (BIT) equipment. Similar efforts are underway (funded by Honeywell) to develop an expert flight control maintenance diagnostic system for fly-by-wire controlled aircraft (such as the F-16 and the X-29). Grumman states that for the X-29, at least, the large amount of flight data needed to build such an expert system is already available. Their third technology thrust is to develop a knowledge-based system for multisensor fusion. Such a system "would accept all the information available from an aircraft's sensors, process that information and make the most relevant and important information available to the pilot at the moment he needs it." It is envisioned that the system would: a) manage the multisensor system, b) detect when a sensor fails, c) schedule sensor use and reconfigure the suite for optimum sensor use, d) assess threats, e) resolve conflicts between sensors, and f) use what is referred to as evidential reasoning to draw conclusions based on the evidence provided.

Grumman also has several year's experience with a combat analysis simulation tool called TOPCAT which can be used to study a broad range of combat and survivability techniques. Each "game" the device plays considers all combinations of initial conditions. The outcome of a particular game is compared with all relevant preceding games, and favorable strategies used in each game are identified and incorporated into each successive game until an "optimum" strategy for winning the game is generated.

Raytheon has several expert system developments underway as well. One goal is to "put a distributed expert system into a deployed radar system" in order to compare a signal with stored Soviet signal data to determine how the incoming signal should be classified. Other goals involve development of systems for missile test stations, battle engagement planning, and target classification. According to their spokesman, Raytheon is cautious not to create AI hype, but rather to demonstrate that expert systems can be useful in operational environments.

Forecasts of Expert Systems Technology

A comprehensive expert system technology forecast is provided by Schindler (1985). He points out that, due to the short supply of qualified personnel in this field, a great amount of activity by researchers is being spent trying to develop the tools and methodologies to be able to simplify the AI product development process. Areas being addressed include the languages, knowledge base schemes, user interfaces, and the inference paradigms. Since hardware costs are also of concern, efforts are underway to address processing speed, code size, and transportability problems between machines. As systems gain more exposure, the ease of use problem is becoming recognized as a most serious area of concern to the AI industry. Many of the systems now under development are directed toward banking, insurance, and investment planning application, rather than to engineering. However, systems using reasoning based on first principles (e.g., laws of physics, chemistry, and mathematics) have also started to emerge.

As noted previously in this report, it is becoming popular to separate the reasoning software (or inference engine) from the domain-specific knowledge base in an expert system, and market the former as "generic" expert system shells into which the user may incorporate his own domain knowledge. A potential drawback of such shell systems is that they still retain enough of the flavor of the original goals for which they were developed, that applications far removed from the original may not be realistically supportable. Some of the more general tools such as the Automated Reasoning Tool (ART), Knowledge Engineering Environment (KEE), and Knowledge Engineering System (KES) may have overcome these problems and are predicted to be more characteristic of expert systems in the mainstream of the future.

Schindler predicts that within the next ten years, custom-made expert systems developed by industry to address particular production and maintenance problems will become the basis of products to be offered commercially. Also, since software availability will limit the utility of AI machines, much effort will be devoted to languages such as OPS83 (by Production Systems, Pittsburgh, PA) which provides constructs similar to those in Pascal that simplify the association between rules and algorithms. Although the OPS83 compiler with 20,000 lines of code written in C can "execute an average of 50 rules per second on a VAX-11-780", the Lisp and Prolog languages will continue to be important tools.

In Schindler's view, automation of the software generation process constitutes what may be the most important application of expert systems. The process control function will probably be the first realization of such capabilities. Another problem that has worried developers is how to extract the knowledge from the domain expert and represent it in the knowledge base. This problem is being addressed by the Boeing Computer Services Co. (Seattle, WA) who will introduce the Expertise Transfer System (ETS) which will allow the development of a knowledge base without the help of a knowledge engineer.

Perhaps the ultimate application of a knowledge base would be to the holdings of active libraries. Optical storage offers the potential medium which could then be queried by an artificial librarian to find any source information desired.

Schindler also suggests that expert system skills will provide the bases for fifth-generation computers in this country. The Japanese approach in their fifth-generation computer project is to rely heavily on massive parallel processing. The U.S. approach (assuming expert system shells as its foundation) relies more heavily on sequential paradigms (though parallel processing will be used as appropriate) since they must step through a search tree.

Combining component and architecture technology, Schindler projects that by 1990 production-type expert systems will run 100 to 200 times faster than they did in 1985. Similarly, the costs of expert systems have dropped dramatically (by a factor of 100) since their first appearance in the mid-1960s. By 1990 a 500-rule system (fairly powerful) may cost as little as $50,000 to develop.

Manuel (1985) argues that if expert systems are to achieve widespread acceptance, they must be able to be used by people who are not AI specialists. The expert system shells (reasoning software or inference engines) that are available run from $200 to $50,000 and can be implemented on several computers, from an Apple Macintosh at the low end to a dedicated and specialized Lisp machine at the high end. However, many of these programs still do not relieve much of the preliminary programming burden.

In reviewing the state-of-the-art, Manuel discusses some of the larger and more comprehensive expert system development tools such as KEE and ART. As of the July 1985 date of that report, IntelliCorp had installed approximately 200 of their KEE systems for about 60 customers (many organizations have up to 10 installations). Similarly, Inference Corporation had installed about 50 copies of their ART system in roughly 30 company facilities. An Inference Corporation executive is quoted as saying that "although ART has extensive on-line help and an abundance of documentation to ease the first-time user along, it requires a learning investment to truly master its range of possibilities." He further claims, however, that ART can be used after a few months of training and some assistance to the customer's programmers to help them build the first one or two systems. NASA is using ART, for example, to create expert systems for space station and shuttle applications. ART is designed to run on specialized Lisp machines (from Symbolics Inc. and Lisp Machine Inc.) but may also be run on VAX computers having VAX-Lisp. According to Manuel, ART consists of four primary elements, including "a knowledge-representation language, a compiler that maps the knowledge into Lisp code and data structures, a run-time applier for solving specific problems, and an environment that allows the developer to control and monitor the system during the development process." Newer 32-bit microprocessor based personal computers are providing a viable hardware alternative to dedicated Lisp machines, although their relative performance is only about half that of the latter. Costs for the dedicated Lisp machines are projected to decline by approximately 35% annually for the next few years, due to fierce competition, VLSI technology, and encroachment by PC machines. The cost of a Lisp machine in 1985 was $55,000 (including two megabytes of RAM, a 140 megabyte disk drive, and a high-resolution monochrome display). At the same time hardware is coming down dramatically in price, sales of AI software are projected to grow from less than $100 million in 1985, to over $1.7 billion in 1990. Many of these programs will be aimed at the PC market to allow potential users to "get their feet wet" and learn by building something useful to them in their specific application arenas. Manuel (1985) describes several relatively inexpensive ($495-$995) packages available for the popular PC machines, as well as some that lay in between sophisticated systems such as ART and the under-$1000 class of programs. The Knowledge Engineering System, for example, can run on several varieties of machines and comes in specialized forms such as the KES.PS, a production oriented, rule-based inference system, KES.Bayes, for development of statistically or probabalistically based inferences, and KES.HT, for hypothesis testing using frame-like descriptions as its knowledge base (e.g., diagnosing engine problems from descriptions of symptoms).

Schwartz (1985) provides an overview of the importance of the PC to the delivery of expert systems in today's market, as well as a projection of the importance of the PC in future expert system developments. He notes the tremendous growth in PC computing power and that 1981 was the year that the total PC and individual workstation computing power equalled that of all other computers combined. When tabulated, 1986 sales of PC's alone are expected to show PC computing power to equal that of all other systems. Although initial versions of expert systems tended to be implemented on dedicated workstations, there is a great trend to use PC's that Schwartz projects will continue and accelerate into the future. The reason for this is the growing availability of AI languages, as well as the large installed base of PC's, together with their declining price. However, he predicts that dedicated workstations will still remain popular due to the reduction in system development time they generally offer.

Schwartz offers a listing of 17 expert system building tools (or shells) available for PC's and written in a variety of languages, including not only the more popular Lisp, Prolog, C, and Turbo Pascal, but also the more traditional Basic, Fortran, and Assembler languages. Although the majority of these systems run on the IBM-PC, some can be implemented on the Apple Macintosh, Commodore 64, Apple II, and Atari 800 machines. Prices for these shells range from a minimum of $20 for a program written in Basic to a maximum of $15,000 for one written in Lisp which can also be linked to other systems. The average price of the programs listed was a little over $3,000. Schwartz predicts that since many of the PC-based systems are aimed at the development of expert systems having less than 200 rules (relatively small), these systems will find widespread use in the business and technical domains as so-called "technician systems." He projects also that little or no help will be required from knowledge engineers due to built-in induction extraction mechanisms in these tools. The systems that will be built will serve to bolster the learning curve and assist the more junior members of an organization in routine operations. In addition, Schwartz predicts that: 1) Dedicated expert systems using application-specific integrated circuits on hand calculator type devices will pervade the world of tomorrow; 2) Many of our daily activities will be aided by low cost expert systems. These include commuter route selection, business and investment decision making, human interaction, personal health care, and even wagering; and 3) Automated knowledge acquisition will allow every expert to become his own knowledge engineer.

Kabrisky (1984) furnishes an especially imaginative, insightful and humorous view of the future promise of AI. As a professor at the Air Force Institute of Technology (AFIT), he describes the attributes of the "Friendly Machine" and projects the verbal exchanges it would have while serving as expert tutor to an elementary school child. The "Friendly Machine" would always know where the child is in the curriculum and how well the child is doing. In Kabrisky's assessment, the verbal exchanges between the "Friendly Machine" and the child require only three components; namely a speech synthesizer, a natural language handler, and a speech recognizer. Since speech synthesizers are already available, and natural language handlers that can deal with constrained but acceptable exchanges have been around for some time (Kabrisky cites Weizenbaum, 1966), only the area of speech recognition remains as the technological barrier to development of the "Friendly Machine." Kabrisky describes progress in his laboratory toward the decoding of natural speech. A single sentence is subjected to an acoustic analysis which provides a listing of probable word sequences. Those word sequences that do not conform to rules of English grammar and sentence structure are rejected, while the remaining sequences become candidates for what was actually meant. The technological problem yet to be overcome involves the machine knowledge base required to properly sort through the list of grammatically acceptable English sentences to find the one that most closely matches the context and semantics provided by the preceding part of the exchange, and the machine's "understanding" of the knowledge of the user. Kabrisky projects that since man is limited in terms of the number of things he can keep track of, the memory and simultaneous manipulation capability of the machine can be expected to provide a problem solving capability that far exceeds our ability to *understand.*

Cross (1984) further describes activities at AFIT's Artificial Intelligence Laboratory and an updated AFIT overview is included in the February 17, 1986 issue of Aviation Week and Space Technology (pg 61). Cross has concentrated on joint applications efforts of interest to DARPA, the operating commands, and other Air Force laboratories. One of these is a prototype interactive mission planning system, developed on a Symbolics Lisp machine. Defense Mapping Agency (DMA) terrain data are used by the system to allow a pilot to preplan a flight course using a mouse and information windows on the display. The program provides a host of outputs, including mission flight time, fuel required, "bingo" fuel point, and known enroute threat locations. A pilot advisory function tells the pilot how a threat may be defeated (e.g., by an increase in airspeed, or a rapid descent). Cross reports very favorable reactions to this demonstration system by Tactical Air Command (TAC) general officers and pilots and hopes to transfer this technology as soon as its maturation and TAC's budget allow.

Basic Research and Philosophical Issues

Winograd, Davis, Dreyfus and Smith (1985) convened as a panel during the Ninth International Joint Conference on Artificial Intelligence to discuss how far expert systems can go in their development and what they can be expected to do. The meeting was spurred by the great wave of enthusiasm about the potential for expert systems in every area of human life and work. Randall Davis (MIT) emphasized that the first question that should be answered does not concern what expert systems can do, but rather what do we know? Once we establish what we know, then, Davis contends that we will be able to address the technology issue and ask ...how easily can we encode that knowledge? Davis further argues that although expert systems is a weak technology, that problem will eventually (though not soon) be remedied. Finally, Davis states that expert systems is a technology for treating ideas that are not completely understood, and that this situation will not change. The implication of these premises is that rule-based expert systems are guaranteed to fail occasionally since they will never have all the knowledge needed to perform flawlessly.

Stuart Dreyfus (University of California, Berkeley), in his presentation to the panel, characterized the development of expert human understanding as progressing through five stages. These he defined as the beginner, advanced beginner, competent performer, proficient performer, and, finally, expert. Dreyfus contends that expert systems can be developed to the competent performer level, at which point behavior is organized "by selecting plans, goals, or perspectives which determine hierarchically what facts to consider and what rules to apply." Going beyond this to the proficiency level requires the performer to recognize the goal based on prior experience in comparable "situations in which goals were chosen and events were chosen and events either confirmed the wisdom of the choice or showed it to be mistaken." The expert, on the other hand, does not need to reason out either strategy or action, but rather simply "associates with each prototypical situation in his memory the decision, action or strategy that he has found to work." Dreyfus indicates that expert systems can be neither proficient nor expert, since they cannot recognize situations holistically without decomposing them into their components, and they cannot "know what to do without applying rules to decompose knowledge." Dreyfus uses the fact that expert systems never perform quite as well as experts to support his contentions.

Brian Smith (Xerox PARC) concentrates on the importance of having an appropriate model in the expert system, which reflects our understanding of the behavior of the system with which we are dealing. He feels we are not lacking in techniques to study the relationships between the system and our model of it (i.e., the activity called program verification). What we lack, however, are tools to study the relationships between the model and the world. Therefore, in Smith's view, we are unable to assess the adequacy of models, or even predict when models might fail. Smith concludes that expert systems should only be developed and used where we have confidence in the accuracy and appropriateness of the underlying model, and that a theory of models needs to be developed to better understand how they should work.

As the final presenter in this panel effort, Terry Winograd (Stanford University) first quoted a participant in the Japanese Fifth Generation Computer Project who had some over-zealous claims for the technology the project may likely produce. He then lamented the term "expert system" because it often promotes an inflated image of how these systems work and what they can do. He stated further that AI program failures will, in part, reflect the user's inability to understand what the program is doing, as opposed to what it appears to be doing if the user accepts the metaphor of "thinking" on the part of the program.

Nilsson (1983), in his Presidential Address at the Annual Meeting of the American Association for Artificial Intelligence, projected that AI, "perhaps together with molecular genetics, will be society's predominant scientific endeavor for the rest of this century and well into the next - just as physics and chemistry predominated during the decades before and after 1900." Although he has a basic research orientation, he urged supporting more applications of AI so that weak spots in the science of AI can be identified and corrected. Nilsson projects that in addition to the present emphasis on expert systems and natural language interfaces, future applications will address AI planning systems such as project planning, error-recovery planning, and robot task planning. In reference to the Japanese Fifth Generation Computer Project, he projects that conceptual breakthroughs will be needed, especially in the areas of common sense reasoning and language processing, if the goals of the project are to be fully met. Finally, Nilsson proposes a new research project that he feels will stimulate advances in the basic science of AI. The goal of this project would be to develop a new class of AI programs that (like time-shared computer operating systems, or airline reservation systems) are never turned off. These programs may be referred to as "computer individuals." Such programs "would have a constantly changing model of the world and of the user(s). They should be able to engage in extended dialogs in natural language." The idea is that such programs would force research in the area of machine learning. Nilsson sees computer individuals performing the roles of personal assistants, meeting schedulers, expert consultants and mobile robots. However, he estimates that we are now at about the same stage in being able to build such programs as physicists were in the 1930's in being able to harness nuclear energy. That is, "we know something profound can be done, we have a few clues about how to proceed, we know that much more basic research must be done - and we want to get on with it."

Demonstrating a more academic orientation, Waltz (1983) provided an assessment of the present status of natural language processing and expert systems. Although most of his concerns in the natural language processing area remain valid, it is interesting to observe that many of his comments regarding problems that need the most work in the expert systems development area are presently enjoying appreciable concentration. (Examples are the ART, KEE, and KES packages described by Schindler (1985) which provide general expert system development capabilities and satisfy Waltz's criticism as to the narrowness of the technology.) Waltz identifies some important areas for further AI research, such as the utility and application of certain forms of inference, including "reasoning by default, reasoning by analogy, synthetic reasoning (i.e., design), and especially planning and reasoning under uncertainty." Another area ripe for research is what Waltz refers to as meta-level architecture, the goal of which is "the construction of programs that can explicitly reason about and control their own problem solving activity." Waltz emphasizes the need for high-quality user interfaces, incorporating friendly features such as high-quality graphics, fast response times and perhaps natural language capabilities, especially for naive users.

Rational Skepticism of AI Claims

Because the people involved in development of these (AI) systems naturally tend to be enthused about the technology they are a part of, the great preponderance of literature available has been from the hands of these developers, or a byproduct of their activities. In any event, the result has been that there are few antagonists that speak with sufficient authority to provide convincing arguments as to why investments in AI technology should not be made. One notable exception is Herbert Dreyfus, author of the book entitled "What Computers Can't Do" (1976). More recently, he has teamed with his brother, Stuart Dreyfus, to produce the book "Mind over Machine" (1986). Both are professors at the University of California, Berkeley, in the philosophy and industrial engineering departments, respectively. In an article in the January 1986 issue of Technology Review, they provide an ultraconservative and disconsenting view of planned DARPA expert system developments. They point to the fact that no expert systems have been developed which display common sense or that can understand natural language, and that "computers are no more able today to deal intelligently with uncertain data than they were a few years ago when our computerized ballistic-missile warning system interpreted radar reflections from a rising moon as an enemy attack." This provides the basis of their major concern that "the success of the DARPA program requires basic scientific breakthroughs, neither the timing nor the nature of which can be predicted." The Dreyfus's position could obviously be the subject of great debate and is reminiscent of the controversy surrounding the Artificial Heart Program at NIH (National Institutes of Health) (see Lane, 1981) where, indeed, knowledge and technology required to make the program successful were not available. In reality, some compromises will doubtless be made in DARPA's Strategic Computing Program, based upon the progression of technology and learning throughout the effort.

Martins (1984) provides a similar, though less caustic, perspective based on his experience (until 1982) as director of the Rand Corp.'s R&D Program in Information Processing Technology and his current position as president of a management consulting group (Intelligent Software Inc., Van Nuys, CA) specializing in applications of advanced computing techniques. Although the power of some of his points may be diluted by the progress made in hardware and software technology since the article was published, his admonitions nevertheless represent a wealth of experience that bear consideration. He relates that even though software development costs are high and development times seem unusually long, the resulting expert systems are "effective only for relatively simple applications," and use code that is "generally hard to understand, debug, extend, and maintain." Furthermore, he states that expert system shells are too expensive, poorly supported and documented, difficult to use, and produce programs that are inefficient and highly limited in their real-world applicability. He further advises that because expert system rules are not independent, adding new rules to a large rule-based system nearly always requires revison of some subset of the existing rules and there is no obvious way of knowing exactly which old rules need to be changed or in what way. Martins attributes the notable expert system successes (most described in Chapter 1 of this report) to six interdependent factors: brilliant programmers, the very narrowly defined and/or easy problems worked, generous funding over a long perod of time, luck, development of customized tools (without the use of shells) and finally, misleading advertising as to the utility and intelligence of some of the programs. He cites the confusion between science and engineering as being the culprit for the above state of affairs. That is, AI researchers, mostly in university laboratories, who should and do "habitually contemplate toy problems, designed to highlight particular themes and issues and to exclude others, produce methods and attitudes appropriate to this kind of scientific investigation (that) are of limited usefulness in confronting difficult, real-world engineering problems." As for the future of expert systems, Martins projects that they will become popular for use with PC's for relatively unsophisticated computational problems such as electronic checklists in the banking, insurance, or sales industries. Developments in relatively more complex areas will rely on "insight, imagination, and deep understanding of both computers and the application domain" which will provide the foundation for future "prodigious achievement."

Lessons Learned

According to the February 17, 1986 issue of Aviation Week and Space Technology (pp 79-81), TRW Inc. (Los Angeles) is attempting to use AI to help solve some of its problems caused by the increasing size and complexity of its computer systems. In so doing, TRW is coming to question some of the popular claims some proponents of AI have made. Some of their experiences are as follows: a) although AI languages provide excellent program writing and debugging capabilities "that can speed programming up by a factor of ten," the programs, once written, "are not much easier to change than conventional languages," since any particular application program is designed to run under such specific conditions that most of the flexibility is lost; b) the notion that computer logic can be developed apart from the knowledge it works on does not hold since the program logic in real applications programs will be affected by the type of knowledge it uses, and vice versa; c) AI programs offer no more reasoning capability than conventional programs using conventional algorithms; d) successful AI programs are usually written by an expert in the problem area having some computer programming capability, not AI specialists, as "common knowledge" would purport, and; e) graphics and symbols have proved easier communication tools than natural language. In one prototype system (BETA -Battlefield Exploitation and Target Acquisition) a variety of battlefield information was collected, collated and displayed to Army officer intelligence analysts in a mock battle evaluation. The analysts could control the amount of information they received from the BETA system. When they relied upon their intuition only, they lost the battle. When they accepted all the information BETA could provide, they still lost the battle. TRW hypothesizes that the ideal amount of information will lie somewhere in between these two extremes and that it will vary for each individual. In addition, since unexpected action is the essence of a good battle, human participation will always be needed.

TRW is also using expert systems to help tune radar identification systems having several thousands of parameters to adjust. Though their original system, developed in the early 1980's with 77 rules, was only 60% effective, its performance has been improved substantially through the addition of more rules (100 now and may grow to 200 in the future). It was found that the original AI language in which the tuner program was written was so slow that it had to be rewritten in a more efficient language (unspecified). TRW's experience also showed that extracting knowledge from experts can be difficult, and since the generalized expert system was written in another language, field representatives will not allow modifications to it for fear of causing software problems they can't fix.

Westinghouse is applying expert system techniques in two major areas, according to spokesmen (see Aviation Week, February 17, 1986, pp 91). One is development of a submersible vehicle that can enter and map a harbor, come back out and compute the optimum locations for mines, and then depart. Another development will use a new AI inference engine processor, which its manufacturer (Zenologic Corp., Berkeley, CA) claims will perform 300,000 logical inferences per second.

Westinghouse found that their first attempt to develop a rule-based expert system to assist them in their in-house overhaul facility did not produce a very efficient or effective product. They then talked to their test technicians to see how they did their jobs and found that nearly every one of them carried a little notebook in their back pockets to write down useful tips based on prior experience. These bits of information were captured into a revised expert system which now interacts with technicians through questions. Westinghouse will next explore how to have the technicians enter logbook type data into the expert system using natural language.

Summary

What has this section provided by way of a glimpse into expert system development activities and capabilities on the near and far horizons? It is obvious that sizeable investments are being made by government agencies, industry, and academia alike in order to develop new hardware, software, and an experience base to advance the technology and state-of-the-art of expert systems at a remarkably rapid rate. Similarly, these developments have spurred considerable philosophical debate and conjecture as to what capabilities future systems may actually be able to demonstrate, as well as what type of academic and technical expertise are most appropriate for advancing the technology most efficiently.

The following listings are provided to assist the reader in formulating an overall impression of where expert systems technology now stands, what applications are under development (some may never reach fruition), what problems and successes are either being experienced or projected, and what are seen as primary needs by the development community.

The Technology

1. Computers especially designed for expert systems applications (with symbolic processors) are being upgraded from original 8-bit (Motorola M-6800) to 32-bit (Motorola M-68020) microprocessors.

2. So-called "systolic processors" (having multiple array processors) are under development (by Carnegie-Mellon University and General Electric Corporation) which will be able to perform 100 million 32-bit arithmetic floating-point calculations per second (i.e., 100 megaflops).

3. Texas Instruments (TI) reported development (under DARPA funding) of a 32-bit Lisp language computer chip, measuring 1 cm. sq. and containing more than 500,000 transistors.

4. TI projects 20 to 30 expert systems could be used on board future aircraft (maybe an upgraded F-16 or Advanced Tactical Fighter - ATF).

5. Expert system shells containing the reasoning software (inference engines) are available that allow the user to incorporate his domain knowledge to build his own expert system on anything from a PC to a sophisticated symbolic processor.

Applications

1. Space station operations.

2. Brilliant munitions (expert systems with pattern recognition capabilities to verify targets).

3. Knowledge-based software engineering/generation.

4. Robotics for production automation.

5. Decision aids.

6. C^3I (Command, Control, Communication and Intelligence) systems.

7. AI test bed for evaluating performance of expert systems which combine FLIR imagery with intelligence information.

8. Nuclear test detection through high-speed seismic sensing and processing.

9. Battlefield management.

10. Tuning of radar identification systems.

11. Pattern recognition algorithm library for aircraft or speech recognition.

12. Maintenance/Diagnostic systems (e.g., flight control system maintenance)

13. In-flight flight control reconfiguration (under battle damage or system failure conditions).

14. Multisensor fusion.

15. Sensor suite reconfiguration (under partial failure conditions).

16. Navigation and navigation system management.

17. Air combat management (multiple ground and airborne threats).

18. Mission planning.

19. Threat radar signal classification.

20. Submersible vehicle for harbor mapping and mine laying.

21. Handling of multiple inflight emergencies.

22. Process control.

23. Library information retrieval systems (requires optical storage medium)

24. Project planning.

Problems

1. Extracting knowledge from experts may be difficult.

2. Software field representatives are reluctant to modify software packages for specific applications for fear of causing problems they cannot solve.

3. Specialized AI languages have not shortened program modification time as anticipated. (However, program development and debugging times are reduced tenfold over conventional languages, according to some authors.)

4. Existing application programs are not easily modified to fit new applications since computer logic cannot be divorced from the specific application being worked.

5. AI programs offer no more reasoning capability than conventional programs using conventional algorithms.

6. Programs cannot be written to display common sense.

7. Programs cannot deal intelligently with uncertain data.

8. Recognition of connected speech and its meaning may prove to be practically (both meanings) insolvable due to contextual uncertainties.

9. Rule-based expert systems are guaranteed to fail occasionally since they will not have all the knowledge they need to perform flawlessly.

10. Expert systems can be expected to perform more poorly than experts, but better than an advanced beginner. As Dreyfus (in Winograd, et. al., 1985) states, going beyond the competent performer level to the expert level requires recognition of situations holistically, without decomposing them into their components, and expert systems cannot do this.

11. The user interface problem will become more severe as more people try to use these systems. The design of intelligent data bases, easy to use and modify software, and good user-system interaction schemes remain major challenges.

12. Expert system shells are overpriced, use code that is difficult to understand, debug and maintain, and are poorly documented. The result is that most programs are inefficient and highly limited in their real-world applicability.

Successes

1. A subject matter expert having some computer programming capability can write successful programs (i.e., you don't have to be an AI specialist).

2. Graphics and symbols communicate better than natural language.

3. Technicians maintaining equipment can provide better diagnostic inputs to an expert system than the engineers who design the equipment.

4. Of the three components needed for natural language discourse (i.e., speech synthesizer, natural language handler, and speech recognizer), only the speech recognition function remains as a technical barrier.

5. Expert systems may handle more information faster than a human.

6. Expert systems will provide the basis for fifth generation computers in this country.

7. The speed of production expert systems will increase dramatically (projected to be 100 to 200 times faster in 1990 than in 1985).

8. Costs of expert systems (both hardware and software) will continue to drop dramatically due to technology and competition.

9. Due to the number of PC installations, PC-based shells have tremendous growth potential in market share of expert system products. (In this country, total PC computing power exceeded that of all other systems in 1986.)

10. Future application-specific expert systems will reside on chips in hand calculator type devices and pervade tomorrow's world.

11. There will be many consumer applications of low-cost expert systems, such as investment decision making, personal health care, and commuter route selection.

12. The knowledge acquisition process will become automated so every expert may become his own knowledge engineer.

13. Given intense exposure to real world problems, together with adequate resources, the lessons learned in AI labs can become the foundation for "prodigious achievement."

Primary Needs

1. Human factors research into user interface problems with expert systems and knowledge acquisition systems is sorely lacking.

2. Expert systems that learn by their own mistakes must be initiated so the AI community can gain some experience with this vital capability.

3. Expert systems and shells that can be used by non-AI specialists and nonprogrammers must be developed if this technology is to reach its full potential.

4. Development of techniques to easily and reliably extract knowledge from the domain specialist is required.

Bibliography

Cross, S.E. (1984). The state-of-the-art in artificial intelligence. In Proceedings of IEEE/AESS Symposium on Aerospace and Electronic Systems, Advanced Concepts and Pioneering Perspectives, Dayton, OH, Sect 4, (pp 1-9), Nov 14-15.

Dreyfus, H. (1979). What Computers Can't Do. Harper & Row Publishers Inc., 10 East 53rd Street, New York, NY, 10022.

Dreyfus, H. and Dreyfus, S. (1986). Why computers may never think like people. Technology Review, 42-61, Jan.

Dreyfus, H. and Dreyfus, S. (1986). Mind Over Machine. The Macmillan Company, 866 Third Ave., New York, NY, 10022.

Kabrisky, M. (1984). The promise in an artificial intelligence future. In Proceedings of IEEE/AESS Symposium on Aerospace and Electronic Systems, Advanced Concepts and Pioneering Perspectives, Dayton, OH, Sect 4, (pp 10-13), Nov 14-15.

Lane, H.W., Beddows, R.G., and Lawrence, P.R. (1981). Managing Large Research and Development Programs. State University of New York Press, State University Plaza, Albany, NY, 12246.

Manuel, T. (1985). The pell-mell rush into expert systems forces integration issue. Electronics, 58, 54-59, Jul 1.

Martins, G.R. (1984). The overselling of expert systems. Datamation, 30, 76-80, Nov 1.

Nilsson, N.J. (1983). Artificial intelligence prepares for 2001. The AI Magazine, 7-14, Winter.

Schindler, M. (1985). 1985 Technology forecast: Expert systems. Electronic Design, 113-134, Jan 10.

Schwartz, T.J. (1985). Artificial intelligence in the personal computer environment today and tomorrow. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Vol 2, (pp 1261-1266), Morgan Kaufmann Publications, Inc., Los Altos, CA 94022.

Technical survey: Artificial intelligence (1986). Aviation Week & Space Technology, 40-92, Feb 17.

Waltz, D. (1983). Artificial intelligence: An assessment of the state of-the-art and recommendation for future directions. The AI Magazine, 55-67, Fall.

Weizenbaum, J. (1966). ELIZA - A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9, (1), 36-45, Jan.

Winograd, T., Davis, R., Dreyfus, S., and Smith, B. (1985). Expert systems: How far can they go? In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Vol 2, (pp 1306-1309), Morgan Kaufman Publishers, Inc., Los Altos, CA 94022.

CHAPTER 4

Human Factors Research in Artificial Intelligence and Expert Systems

Introduction

To be discussed in this section are human factors design issues and research relating to the development of expert systems for military and/or industrial applications. For those interested in a slightly broader perspective on human factors and AI (i.e., including intelligent robotics, and natural-language understanding, as well as expert systems) the review by Hillman (1985) is highly recommended.

The January 1986 issue of IEEE Spectrum (pp 86) quoted Andrew P. Sage (President of IEEE's Systems, Man, and Cybernetics Society) as saying that "the major challenge in this (expert system) field is to design systems for easier human interaction." Sage elaborated by saying that "the major breakthroughs needed to solve the human interaction problems depend on design of intelligent data bases, behavioral and human factors in systems design, and intelligent programming and decision-support generators."

The intent here is to inspect in some detail, those studies that were identified through a review of the literature to address important design characteristics of expert systems and to illustrate the direction of research activity that, although generated in related study areas, can nevertheless be brought to bear on the subject.

Design Considerations for Expert Systems

Although in the review of literature for this effort there were few citations identified that specifically treated the human factors design aspects of the human-expert system interface, there is a growing body of literature that treats human-computer interaction, some of which may be applied to user interfaces for expert systems. A significant portion of that literature can be found in the annual "Computer-Human Interface Proceedings" published by the ACM (Association for Computing Machinery). In one study from this source (Galambos, 1983), the nature of keyboard entry errors made by novices as they learned to use their first text editor provides an insightful projection of what might happen as novices may try to interact with their first expert system, if that interaction could only be performed through an unfamiliar set of keyboard commands. The challenge in designing any such system is to try to minimize these painful early errors so that the systems can be made to perform as designed as soon as possible to gain rapid user acceptance and productivity.

Kidd (1985) describes three problem areas in the construction and use of expert systems. These pertain specifically to: a) the difficulty of eliciting knowledge from the expert, b) how one represents the knowledge base within the system, both in terms of adequacy and intelligibility to the user, and c) designing the user interface so that a variety of user needs can be satisfied. Kidd concludes that for systems to be successful in field applications, they must be able to be regarded as consultants, rather than as purely domain specialists. The orientation in the DARPA funded "Pilot's Associate Program" seems to reflect this same consultative nature of the expert systems to be developed.

One of the more systematic attempts to explore how design principles for human-computer interfaces might be developed is described by Norman (1983). In his "User Centered System Design" project, Norman characterizes the sort of tradeoff analysis that he feels must provide the basis for any particular design decision. He states that one of the basic tradeoffs pervading many design issues is that "factors that increase informativeness tend to decrease the amount of available workspace and system responsiveness." That is, any display may have anywhere from too much to too little explanatory information available, depending on the experience of the user. The more complete the display, the more time required to generate it, and the less workspace will be available. In the extreme case, some commercial vendors have learned that when they try to make their systems very easy to use with little or no experience, the associated processing overhead degrades system response time to such an extent that serious user complaints have resulted.* In any event, it is clear that there is an important tradeoff between how much supportive information should be provided, the impact that has on remaining available workspace, and the acceptable amount of time required for processing. In addition, this tradeoff varies as a function of recency and amount of user experience. Norman advocates using a psychological measure of "User Satisfaction" to quantify the tradeoff relationships among these parameters so as to be able to determine when user satisfaction has been maximized to a reasonable extent. It would seem mandatory to include a similar capability in airborne expert systems so that the aircrewmember could customize processing overhead and display update rates according to his needs and experience, thereby maximizing satisfaction and, hopefully, man-machine performance.

Rouse (1984) describes an integrated methodology for a top-down approach to the development and evaluation of decision support systems (DSS's). Since expert systems seem to have DSS's as first cousins, the highly formalized procedures presented by Rouse may produce beneficial effects when applied to expert system development efforts. In particular, the design of information provided to the user, as a function of the system's model of the user, may be facilitated considerably using Rouse's approach. According to Rouse, requirements

* As Norman states, ".. the long, informative displays or sequence of questions, options, or menus that may make a system usable by the beginner are disruptive to the expert who knows exactly what action is to be specified and wishes to minimize the time and mental effort required to do the specification."

on the design of information presented can be expected to be markedly different as a function of whether the decision making task involves situation assessment, as opposed to planning and commitment, or execution and monitoring. Similarly, the decision making strategy will differ as a function of the familiarity of the situation confronting the user. For example, familiar situations allow the user to call up appropriate courses of action directly. Therefore, the information should be presented such that the familiar pattern may be recognized. On the other hand, totally unfamiliar situations require the user to employ analytical reasoning abilities. Information to support decisions in these situations should emphasize causal or functional relationships. Rouse describes various levels of evaluation for DSS's. For example, the compatibility between information displayed to the user, and responses required from the user must be assured; the "messages" displayed by the system must be understandable by the user; and the DSS must actually improve man-machine performance. Such evaluations may be performed analytically or empirically. Details on a taxonomy of knowledge requirements for decision makers to be able to use a DSS are provided in Rouse, et al. (1984).

The System and Users Need Models of Each Other

Another potentially important aspect of efficient and proficient interaction between the expert system and the user is that each has an appropriate model of the other. Since both the human and the expert system may act as decision makers (i.e., either independently or jointly), the allocation of tasks to each side of the interface is a basic system design issue. Ideally, a particular task should be allocated to the machine or the human, depending on which has the resources available at that moment to do the job best. Revesman and Greenstein (1983) delineate the development and application of a computer model to predict actions a human will take at a given point in time within a process control situation. The model has two stages, representing human event detection performance (based on a discriminant analysis*) and control action prediction, respectively. The model was applied to a situation in which simulated variations of sheet metal thicknesses being produced by 9 separate machines (i.e. as represented by a set of 9 traces on a CRT) had to be monitored by subjects to make a decision as to when a machine needed to be repaired. Information as to mean time between failures (MTBF - in seconds for each machine), time to repair (TTR) the machine (TTR - in seconds), and the cost per second of allowing the machine to remain failed was displayed above each trace. Subjects could effect a "repair" by pushing a numeric key associated with each machine. Both the subjects and the model used the MTBF, TTR and cost information to make the decision to "repair the machines." In one condition, the model performed all repairs based on its knowledge of what the human would do at that time to attempt to minimize the expected cost of operating the system. No actions were made which conflicted with or duplicated those of the subject. In another condition, no model of the subject was used and the computer simply made repairs on its own so as to minimize the expected cost of operating the system. For each of the above conditions, two levels of computer communication were used in which the computer either did or did not inform the subject when an action was taken. The results of this study indicated that system performance improved when the computer had a model of the human and the two were operating in parallel. Communication to the human of what the model was doing further enhanced system performance.

Along similar lines of concern, Harris and Helander (1984) raise several issues regarding ergonomic aspects of human-machine information exchange and the symbiotic (see Licklider, 1960) process. They maintain that since it may be impossible to determine which member of the human-

---

* Discriminant analysis is a mathematical means of maximizing the distance in multifactorial space between the weighted parameter representations of two objects. For example, we might ask how dogs and cats could be best parametrically described so as to always be able to discriminate between them.

machine dyad might be at fault when an incorrect output is produced (because the behavior of each is partly dependent on the other), a whole new arena of product liability will be forged with respect to misleading decision aids. The authors illustrate their point by hypothesizing the range of acceptance of an expert system's output as a function of the experience level of the user. For example, the recommendations of a MYCIN-like system would be regarded differently by a highly experienced physician than by a fresh paramedic at a remote site who may have no other sources of reference available. As another example, the output of a weather advisory system for general aviation aircraft may lead to different levels of acceptance by users based on the interaction of the apparent authoritativeness of the system and the differential personality tendencies of the users, as well as their experience level. In any event, as these authors state, "the influence of authoritative demeanor (on the part of the expert system) on human decision-making is yet unknown." Harris and Helander also differentiate between simulated versus synthetic intelligence and the resulting effect on how system failure would be viewed by their users. Synthetic intelligence involves the notion that since machines are not biological, they need not be constrained in their intellectual activity by what we know of as biological intelligence. New approaches to problem solving by machines employing synthetic intelligence may be vastly different in their functioning and information processing approaches. Since a system designed with simulated intelligence would presumably behave as closely to the human as possible, the failure modes of this type system should be more predictable but may be less easily detected than their synthetic intelligence counterparts.

Harris and Helander's final point has to do specifically with the requirement to develop and maintain an adequate model of the user throughout the human-machine dialogue. They discuss the difficulty of finding engineering documentation that specifies the physical and psychological characteristics of the user. When such models have been developed, their application specificity has disallowed identification of general principles for their design. These authors conclude that a great deal of maturation will be required for psychological theory to be helpful to systems engineering problems. The concerns expressed by these authors are especially relevant to the development of expert systems for cockpit applications. Surely the information packaging and content will need to be varied as a function of both the general experience of the pilot, and his/her experience with the particular expert system being used. The requirements for development of a user model could be minimized, however, by the use of a credit card type user identifier and system conditioning capability whereby major configurations of the onboard expert systems would be dictated by the known attributes of the pilot. Such individualized conditioning capabilities are popular topics for those interested in advanced aircraft control systems as well (see Reising and Moss, 1985).

Evaluation of User Performance

The first impression offered by any expert system, as well as its continued utility, is a function of the design of the user interface. Whiteside et al. (1985) evaluated user performance with command, menu, and iconic interfaces across seven commercially developed, top-level interactive software systems (four command, one menu, and two iconic). A population of 76 subjects was used which represented three classes of computer experience, ranging from little or no computer experience, to transfer users (i.e., persons who used interactive computers daily but had no experience with the particular system being evaluated), to daily users of the system being tested. After a brief introduction to the system to be used (system documentation was also provided), subjects were asked to perform file manipulation tasks involving simple operations such as displaying, merging, and sending files to other users. Performance was evaluated objectively through a composite user-performance score which combined task completion times, proportion of tasks completed, and a 5-minute criterion time interval to provide a measure of the rate of tasks completed per 5-minute interval. (This interval represented a presumed mean fastest possible time for a practiced subject.) Sessions were also videotaped and interviews and questionnaires were administered at the completion of the experiment. The results of this study highlight the complexity of the user interface problem. Although there were found to be large usability differences between the systems tested, these differences were not related to the particular style of interface, but rather to particular design qualities built into the interfaces, regardless of what style they were. There were also problems common to all systems. For example, there were difficulties concerning lack of feedback on all systems. Users on all systems were found to repeat previously made entries, being apparently unaware that their input had already been accepted. They also consistently overlooked messages that were displayed anywhere but in the center of the screen. When problems were broken down by style of interface, it was found that many problems with the iconic systems related to the required rote knowledge of the complex syntax of mouse position, number of clicks, timing of clicks, up-stroke vs. down-stroke, and choice of button pressing while moving vs. pressing while not moving. A major problem with the command systems was the lack of examples of command-like syntax from which to fashion one's own commands. Users were also overwhelmed by the amount of help available (which was sometimes irrelevant) as well as error messages, many of which were misleading and confusing. Many users of the menu system "became so involved with maneuvering through the menu structure that by the time they had done this successfully, they forgot what the task was" ..and they.. "got stuck for want of information about how to change states." In fact, the menu system was found to produce the poorest performance and subjective reactions, a surprising result since the primary intent of a menu driven system is to facilitate the user interface. Remarkably enough, the best systems (which represented at least one of each style of system) were the best for new users and system users alike. This result argues strongly against the existence of a learnability versus usability tradeoff for these types of systems and instead supports the notion that these qualities are congruent.

Roberts and Moran (1983) reported similar findings in their evaluation of text editors. Whiteside et al. conclude that it is the crafting of these interfaces that is most important and that this crafting comes with product maturity. They feel, however, that the crafting process can be accelerated through application of human factors expertise and iterative testing of prototype software prior to final release of the product.

Regarding another mode of potential interface, Michaelis, Miller, and Hendler (1982) describe efforts by a team composed of artificial intelligence and human factors specialists to develop a computer-processable, human-engineered subset of natural language which could be used to teach computer programming in an "intelligent tutoring system". The impetus for their work is based on research by Ford, Weeks, and Chapanis (1980) and Michaelis (1980) in which one member of each of several two-person teams received instruction from his/her partner regarding how to build an odd shaped wooden model from an assemblage of pieces provided. The instructions were provided either by voice or by teletypewriters (Ford et al.) or by teletypewriters alone (Michaelis). In each study, half the subjects were rewarded for minimizing the number of words used to reach a correct solution (i.e., the fewer words used, the greater their earnings), while the other half was rewarded only on the occurrence of a correct solution. Results of these studies showed that not only did self-imposed brevity not affect problem solving accuracy, but also self-limited teams actually solved their problems faster than their unrestricted counterparts and they did so using about one fifth as many words with about one third as many messages and word types. Next, the protocols (recorded transactions) were analyzed to determine if there were systematic differences in problem-solving strategies or order of subtasks performed. None were found. These studies provide strong evidence that asking people to be concise in their communications with the attendant reduction and restriction in natural language usage may improve, not hamper, the communication process.

In a similar study, Kelly and Chapanis (1977) found that when the communication process was limited to only the 300 words most often used by an unrestricted team, the problem solving accuracy and speed by restricted versus unrestricted teams did not differ.

In total, the above studies imply that constrained natural language communication between a pilot and the on-board expert systems could be developed that would not result in diminished understanding by the pilot or by the expert systems of the knowledge, intentions, or actions of each.

Jenkins (1984) relates an experiment in which an expert system (in the form of a response tree) was used to assist in the solution of a process control problem in a simulated nuclear power plant application. Specifically, subjects were provided training on the types of response trees they would be required to use to identify the best solution to a reactor core cooling problem. Eight groups of subjects (in a 2 X 4 design) corresponded to two levels of task difficulty, and four levels of use of the expert system, ranging from mandated use, to optional use, to knowledge of how to use the system, but it was inoperable, to no knowledge or use of the expert system whatsoever. Dependent measures included maximum fuel temperature reached, number of control activations, and number of response tree uses. The results provide several interesting insights into the interpretation and use of response trees. As might be expected, when offered a choice, the response trees tended to be used only on the more difficult problems, and, when used, caused considerably more time to be spent to reach problem solution than when the response trees were not available. Not only did subjects tend not to use the response trees when they were available, their use did not improve operator performance. The authors state that these results may have occurred because the test scenarios may have been too easy, or the simulated cooling system may not have been complex enough. Also, even though it had been emphasized to operators during their training period that failures had to be manually inputted to the response tree aid, their performance under pressure indicated that they presumed the computer already knew about the failures. Needless to say, this mismatch between computer capabilities and user expectation reduced the usefulness of the aids and again highlights the requirement for an appropriate model by each side of the user-system interface. A final and very important lesson offered by this research has to do with the credibility of the information provided to the user on his/her display. It was found that even when the displayed information was incorrect (because of incorrect user inputs to the computer), the operators still placed high confidence in the accuracy of the computer generated data.

Jenkins's findings emphasize the importance of aligning the user's expectations of the performance of the expert system with its actual capabilities. Also, displays must incorporate sufficient information for the operator to detect immediately when he/she has made an irrational or incoherent input. This does not necessarily mean that the system must be able to detect every inconsistency, but rather that a sufficient record of action sequences be represented for the operator to detect when a blunder has been made. The challenge to the expert system design community will be to provide this capability while still adhering to criteria of display simplicity and minimal clutter. Reliance on graphic and pictorial representations will help accomplish these objectives.

Summary

Where do we stand regarding the development of a human factors data base for expert systems? As has been attempted to be shown here, many of our early grapplings have come from the allied research areas of human-computer interaction, decision support systems, and response (or fault) tree diagnostic systems. It is only fitting that we extract usable information from these sister domains where appropriate. However, it is also necessary that the human factors community establish the required contacts with developers of expert systems so that each discipline is made aware of the needs and knowledge base of the other. Cooperative development of future expert systems could be expected to facilitate the development process as well as result in systems that are easier to use and market.

Prime candidates for cooperative research and development efforts by human factors and expert systems specialists include the following activity areas:

a) Extraction of knowledge from domain specialists.

b) Exploring alternative interfacing strategies such as the use of natural language.

c) Developing models of expert system behavior that communicate immediately to the user what the system is doing and why.

d) Developing display techniques that facilitate the above communication process.

e) Developing user activity definitions that convey satisfactorily to the system what the user is doing and why. (The question here is how to infer intent through a sequence of actions.)

f) Developing expert system building software that allows a domain specialist (who is not a computer scientist) to most easily construct an expert system for his own area.

g) Cooperative development of research goals and protocols that better address problems of utility assessment and user satisfaction.

Bibliography

Ford, W.R. Weeks, G.D., and Chapanis, A. (1980). The effect of self-imposed brevity on the structure of diadic communication. The Journal of Psychology, 104, 87-103.

Galambos, J.A. Wikler, E.S., Black, J.B., and Sebrechts, M.M. (1983). How to tell your computer what you mean: Ostension in interactive systems. In Proceedings of the ACM, (pp 182-185), Dec.

Harris, S.D. and Helander, M.G. (1984). Machine intelligence in real systems: Some ergonomics issues. In Proceedings of the First USA-Japan Conference on Human-Computer Interaction. G. Salvendy (Ed.) (pp 267-277). Elsevier Science Publishers, Amsterdam, The Netherlands.

Hillman, D.J. (1985). Artificial Intelligence. Human Factors, 27 (1), 21-31.

Jenkins, J.P. (1984). An application of an expert system to problem solving in process control displays. In Proceedings of the First USA-Japan Conference on Human-Computer Interaction. G. Salvendy (Ed.) (pp 255-260). Elsevier Science Publishers, Amsterdam, The Netherlands.

Kelly, M.J. and Chapanis, A. (1977) Limited vocabulary natural language dialogue. International Journal of Man-Machine Studies, 9, 479-501.

Kidd, A. (1985). Human factors in expert systems. Data Processing, 27 (4), 15-17.

Licklider, J. (1960). Man-computer symbiosis. In IRE Transactions of Human Factors in Electronics, (pp 4-11).

Longuet-Higgins, M. (1981). A new theoretical psychology? Cognition, 10, 197-200.

Michaelis, P.R. (1980). Cooperative problem solving by like- and mixed-sex teams in a teletypewriter mode with unlimited, self-limited, introduced and anonymous conditions. JSAS Catalog of Selected Documents in Psychology, 10, 35-36 (Ms. No. 2066).

Michaelis, P.R., Miller, M.L., and Hendler, J.A. (1982). Artificial intelligence and human factors engineering: A necessary synergism in the interface of the future. In A. Badre and B. Shneiderman (Eds.), Directions in Human/Computer Interaction, (pp 79-94), Ablex Publishing, Norwood, NJ 07648.

Norman, D.A. (1983). Design principles for human-computer interfaces. In Proceedings of the ACM, Computer-Human Interface, (pp 1-10), Dec.

Reising, J.M. and Moss, R.W. (1985). 2010: The symbionic cockpit. In Proceedings of NAECON, (pp 1050-1054), May.

Revesman, M.E. and Greenstein, J.S. (1983). Application of a model of human decision making for human/computer communication. In Proceedings of the ACM, Computer-Human Interface, (pp 107-111), Dec.

Roberts, T.L. and Moran, T.D. (1983). The evaluation of text editors: Methodology and empirical results. In Proceedings of the ACM, Communications, (pp 265-283), Apr.

Rouse, W.B. (1984). Design and evaluation of computer-based decision support systems. In Proceedings of the First USA-Japan Conference on Human-Computer Interactions. G. Salvendy (Ed.) (pp 229-246). Elsevier Publishers, Amsterdam, The Netherlands.

Rouse, W.B. and Rouse, S.H. (1984). Human information seeking and design of information systems. Info. Proc. and Mgt, 20 (1), 129-138.

The Specialities: Knowledge Support Systems Design (1986). IEEE Spectrum, 86, Jan.

Whiteside, J., Jones, S., Levy, P.S., and Wixon, D. (1985). User performance with command, menu, and iconic interfaces. In Proceedings of the ACM, Computer-Human Interface, (pp 185-190), Apr.

CHAPTER 5

Image Understanding

Introduction

Because of the potential merging of expert system capabilities into pattern recognition and image understanding subsystems in future airborne applications, it is important to review the technology and research being performed in the area variously referred to as computer vision, visual sensing, image understanding, and machine pattern recognition. Development and application of robotic vision systems, per se, is discussed in the Handbook of Artificial Intelligence, Vol III (Cohen and Feigenbaum, 1982) and will not be reviewed further here. The same volume contains a description of three computer vision systems (pp 306-312), each developed during the 1970's, which have provided the historical foundation for development efforts today.

Horn (1980) states that the task of a vision system is to produce a description of what is being viewed. The input to this process may be one or more images in the form of two-dimensional distributions of scene radiance values. The output must be able to specify aspects of the three-dimensional reality, as well as be useful to perform some specific task.

Horn feels that machine vision is complex enough to warrant its own methodology and will not likely succumb to the traditional bag of tricks such as linear systems theory, statistics or communications theory. Systems can be dichotomized into those which work at reasonable speeds, and those which work reasonably well. An example of the former are machines that extract topographical information from stereo pairs of aerial photos. They work reasonably fast but still require human intervention when gross confusion occurs due to differences in, for instance, sun angle or slope shading. So called "automatic" terrain classification systems have also been built that suffer the same problems. On the other hand, MIT researchers have developed usable edge detection and line finding systems that work well but require large capacity computers and long processing times.

Horn describes the roots of machine vision as coming from the fields of image processing, pattern recognition and scene analysis. Image processing typically employs transform techniques to result in an image for human viewing that is better than the original. Pattern recognition involves mathematical feature extraction, followed by pattern classification and has not lived up to the promises implied by the level of mathematical sophistication typically employed. Scene analysis efforts attempt to produce line drawings of the essential elements of the scene in order to determine its contents. Technical problems in performing useful scene analysis have led some researchers to exploit prior knowledge regarding the likely contents of the scene being viewed. The direction of research in image understanding is shifting from an analysis of relatively rapidly changing or uniform intensity areas to analyses of such things as reflectance color, shape, illumination conditions, and shadowing.

Processing Architecture

Mudd (1980) describes work performed by Hughes, Malibu under the DARPA Image Understanding Program to determine processing requirements, as well as machine organization for complex image processing and analysis. The intended application for this work is both the real-time tactical scenario requiring remote sensing and analysis, as well as automated prediction and inspection for machine assembly.

Although the complexity of the image understanding problem, in terms of the processing requirements, algorithm definition and throughput speed have not allowed a unique solution, technology developments in the Very Large Scale Integrated circuit (VLSI) and Very High Speed

Integrated circuit (VHSI) areas are predicted to provide the salvation needed. Estimates of processing requirements for various manipulations (in terms of millions of instructions per second, or MIPS) were orders of magnitude beyond the capabilities of machines current at the 1980 publication time of this report.

Mudd projects that the manipulations that need to be made would be by so-called "primitive" processors, each performing a single function such as convolution, edge detection, histogramming, or other statistical operations. The output of these primitives would be passed to a symbolic processor which would match features, determine the shape of objects, and make mission related decisions. Many of the primitives themselves are envisioned to be incorporated directly at the focal plane of the sensor. Charge coupled devices (CCD's) and metal-oxide-semiconductor (MOS) materials technology offer significant opportunities for such integration, since CCD's (imagers themselves) and MOS chips can be packaged as sensing/processing units. Mudd contends that with such technology, together with the development of other low level primitives, processing at real-time television rates is possible.

Finally, control and integration of primitives is considered since each primitive would operate at its own rate and have unique connections and timing requirements with other processors. The anticipated gate densities for VLSI and VHSI components (on the order of 10^5/chip and clock rates near 10^8 Hz) will allow much of the hardware to be put on relatively few chips.

Model Developments

In his book "Artificial Intelligence," Winston (1984)* discusses some of the problems involved in making a computer understand an image.

---

* Patrick Henry Winston is a professor of computer science and Director of the Artificial Intelligence Laboratory at MIT. He states that most of the ideas presented in his book were developed through long-sustained support by DARPA and ONR (Office of Naval Research) and that it has been personnel in these agencies who made the field of AI possible.

Although his orientation is toward robot vision systems, the ideas presented are equally applicable to an airborne sensor and processing system. In his approach, the first consideration in image understanding is to map image brightness (more properly termed image irradiance, or the power per unit area at the sensor plane). Facts about brightness changes are then used to form a "primal sketch" (a term coined by Marr, 1982) which is devoid of any surface texture information and contains only an outline of the major surfaces of the object. From the primal sketch, a "2½-D sketch" is constructed which provides arrows emanating from the surfaces depicting vectors normal to those surfaces. The final level of representation explicitly defines how objects fill space and is referred to as the "world model." The body of a conventional desk-top telephone, for example, can be thought of as a combination of a vaguely wedge-shaped object having two U-shaped protrusions, all of which occupy some finite space. Winston hypothesizes that "all powerful vision systems must use something like primal sketches, 2½-D sketches, and world models." However, he points out that no one knows how this information is organized or flows through the various representations, as it must in the human (ostensibly, *prior to organization by any* Gestalt types of operations). He suggests that in a top-down flow, for example, our image understanding may be based on "controlled hallucination... whereby early vision is guided by firm expectations about what is to be seen"... and ... "in this respect, image understanding is like language understanding." In any event, Winston asserts that it is clear that visual systems will use whatever information is best at any point in time and that they are sensitive to and use information quality (image quality) in the decision making process.

In Winston's treatment, the next topic in the image understanding process is how one localizes edges in a scene. One way to deal with typically noisy edges in an image is to construct an average-brightness array which effectively smooths small differences in brightness. If one then constructs an average-difference array (i.e., differences between adjacent pixels) from the average-brightness array, and then performs that process a second time, the result is called a point-spread function (two-dimensional in this case). This function identifies how a single isolated brightness point input to a sensor will spread across an output image. There is physiological evidence (Hubel and Wiesel, 1962, 1979) that, at the retinal ganglion and visual cortex levels, there are cells having response characteristics that mimic this function. From a two-dimensional perspective, this function resembles a Mexican hat, thus its designation as a sombrero filter. Hubel and Wiesel found corresponding spatial excitatory and inhibitory effects in terms of modulation of cell firing rate about spontaneous levels.

The perception of distance from stereo cues also must be accounted for by an image understanding system. Winston points out that the fundamental problem in stereo vision is correlating objects in the left and right images so that their disparity can be assessed. Exactly how a biological visual system does this is not well understood, but it is clear that any machine visual system would have to perform this function if it is expected to move in or interact with its environment.

The final area discussed by Winston that will be covered here has to do with the determination of surface orientation from shading information. An image can be considered to be a reflectance map of the objects contained within it. The computational trick is to determine object orientation based on known information concerning the relative position of the light source, the visual system, and reflectance characteristics of the object's surface. If one were to illuminate a sphere from various angles relative to the observer, for example, isobrightness lines could be determined that relate to the direction of the illuminating source according to the cosine of the angle between the observer and the source. Such information can be used by biological and machine vision systems alike to determine the orientation of objects when the object geometry and position of the illuminant is known.

Another excellent report that addresses more recent developments in computer vision and pattern recognition is provided by Fu and Rosenfeld

(1984). Since remote sensing typically yields only a 2-D image, image analysis is often only two-dimensional. Their treatment of computer vision parallels that of Winston (1984) in his discussion of image understanding and will not be covered further. However, Fu and Rosenfeld define pattern recognition to be concerned primarily with the description and analysis of measurements taken from physical or mental processes, and indicate that all mathematical methods that have thus far been used to solve pattern recognition problems can be categorized as either decision theoretic (i.e., statistical) or syntactic (i.e., structural). Decision theoretic approaches represent patterns in terms of N features (e.g., shape, texture, spectral components) represented in an N-dimensional feature vector, which is then compared with some criterion to arrive at a measure of similarity, distance, a likelihood function, or a discriminant function. Feature space is transformed so that an optimization criterion may be applied to discriminate among classes of objects so as to either maximize the distance between classes, minimize distance within a class, or perhaps both. Popular approaches (with digitized scene pixels) for generating pattern features include the Fourier, Harr, and Walsh-Hadamard transformations. To reduce the number of dimensions in feature space, the Karhunen-Loeve expansion and/or the principal components method may be employed.

A second approach to the categorization of a set of N features is to use a distance measure that relates to the probability of misrecognition so that features may be selected which maximize the distance between the subset of features selected to classify the objects. A Bayes classification rule and/or contextual information is often used to minimize the probability of misrecognition.

Application of the syntactic approach requires that the pattern be represented as a pattern of primitives and the relations between them, or in the form of a string or tree. Primitives are the simplest subpatterns or components that can be easily treated using decision-theoretic techniques. Patterns between the primitives are represented by a sentence (consisting of a string or a tree) using a specialized pattern description language having a formal pattern grammar. Typical primitives for handwriting may be various types of strokes, while those for continuous speech may be phonemes. However, since neither of these types of primitives can be extracted easily by a machine, recognition of handwriting, as well as continuous speech, remains a significant challenge.

Syntactic recognition may use anything from simple template matching to a full syntactic analysis (including grammatical descriptions or parsing) to generate a complete description of the pattern and the syntactical relationships involved.

Fu and Rosenfeld conclude that the most serious problems for pattern recognition technology involve identification of efficient feature extraction and selection techniques*, together with the selection and extraction of primitives. Also, since pattern recognition algorithms tend to be computationally slow, they await special computer architectures and VLSI technology to provide support to real-time applications.

Barrow and Tenenbaum (1982) propose a computer model by which line drawings may be interpreted as three-dimensional surfaces. They base their model on the presumption that sufficient information as to surface orientation is available from extremal (object edge) and discontinuity (superposition) boundaries so that three-dimensional space curves and smooth surfaces within the scene may be logically reconstructed. Conceptually, the model behaves as most people do when asked to interpret, in three dimensions, what is presented to them in two. For example, an elipse is most commonly interpreted as a tilted circle, rather than "some bizarrely twisting curve (or even a discontinuous one)

---

* The same may be said for speech recognition.

that has the same image." The extremal boundaries define points in the image where the surface orientation can be inferred exactly (i.e., normal to the line-of-sight, or perpendicular to the line tangent to the extremal boundary). Although discontinuity boundaries do not directly constrain possible surface orientation, "local two-dimensional curvature in the image does provide a statistical constraint on the local plane of the corresponding three-dimensional space curve, and thus relative depth along the curve." Since surface normals at each boundary must be orthogonal to the tangent at any point on the surface, the only remaining degree of freedom for surface orientation is a swing along the axis of the tangent line.

The Barrow and Tenenbaum model consists of three steps. First, each line in the image must be sorted as to whether it represents an extremal or discontinuity boundary. Secondly, normals are constructed along extremal boundaries and surface contours are interpreted on the basis of logical geometry. Lastly, boundary conditions are used as the basis for interpolation to construct surface orientations. The model has been tested using simple geometric shapes (e.g., a trapezoid was correctly interpreted as a tilted rectangle, and an elipse was correctly interpreted as a tilted circle). However, when less than ideal (i.e., noisy) data are input to the model, the computational procedure becomes slow and ineffective. Another limitation to the approach involves the problem of how to extract line (boundary) information from grey-level imagery. Nonetheless, Barrow and Tenenbaum contend that a developmental approach such as theirs is significant since it can be used to explain surface perception without having to resort to analytic photometry techniques.

Some Applications

Bajcsy, Joshi, Krotkov, and Zwarico (1985) describe a basic research project to develop a prototype system (LandScan-Language Driven Scene Analysis) having data-driven modules which build a surface graph of an aerial scene from stereo images. A set of query-driven modules allow the user to probe the data base for information of interest. A scene model is then constructed consisting of those objects that represent the interests of the user. Though not yet developed, the authors intend to provide a natural language interface for the query portion of the system which would "apply locative linguistic constructs to some representation of visual data and reason about this data."

Tenenbaum, Barrow, Bolles, Fischler, and Wolf (1980) discuss experiments dealing with the extraction of information from LANDSAT type imagery for monitoring and/or tracking conditions at geographic sites. Applications may include industrial plants where thermal or chemical pollutants, oil spillage, etc., may be of interest, or the detection and tracking of forest fires or icebergs. The first task in their procedure is to determine the geometric correspondence between the sensor image and map coordinates. A camera model is developed that specifies the location, orientation, and focal length of the camera, and trigonometry is applied to evolve the solution. A map data base is used to provide a 3-D description of major landmarks and monitoring sites: the San Francisco Bay Area in these experiments. The camera model is used to predict the appearance of landmarks (both natural and manmade) and slight differences between predicted and actual appearances are used to calibrate the camera model further. Once final calibration has been achieved, the authors claim it is possible to determine the precise location of geographic sites to within one pixel. (There are 16 million pixels in a typical 4000 X 4000 LANDSAT image.) This allows one to answer such questions as whether a plant has increased its discharge of pollutants, or whether sea water has intruded further up a river delta. Such data provide a much richer base to draw from than is possible using the traditional multispectral analysis approach and the resulting image statistics. Similarly, several types of monitoring tasks become trivial. The counting of box cars on a railway, ships in a harbor, planes on a runway or cars on a highway are examples.

Tenenbaum, et al. (1980) conjecture that the payoffs from automated monitoring using this technology could be substantial. They envision a system that would extract updated site information from new imagery and periodically distribute it on a subscription basis to interested users via communication satellites. An expert system using such information could provide substantial intelligence input to tactical or strategic planning operations.

In what is becoming a classic paper, Sakai, Nagai, and Kanade (1972) present a processing scheme in which a binary-valued (thresholded) line-like picture is generated from a digitized (140 X 208 pixels, 32 gray shades) photo. This scheme uses a Laplacian operator (i.e., a two-dimensional secondary differentiation) to identify areas where brightness changes are high. The positions of facial features (e.g., eyes, nose, mouth, chin, and chin contour) are defined in the line picture. The program considers predictive contextual (or syntactic) information to assist in locating the geometric features. For example, it looks for eyes where eyes should be, not where the mouth might be. The procedures for finding the various facial parts are divided into subroutines and these may be automatically modified to do more detailed analysis if the program fails to detect the part in the predicted area on the first pass. The general technique used to identify the position of facial parts may seem primitive by today's (1986) standards. It is done through what is referred to as "integral projection" and conceptually amounts to placing a slit over the feature suspected to be a nose, eye, etc., and counting the number of digital pixels across the slit. A horizontal slit is used across the eye areas, and a vertical slit is used on the nose/mouth/chin area. Histogram plots of actual vs. predicted, one-dimensional pixel distributions are compared, and the sampling slit position is readjusted iteratively until an acceptably close statistical match is made. A chin contour is similarly located using slits along radial projections from the center of the upper lip.

The success rate across some 800 photographs of people of various ages, with and without hats, beards, and glasses and at various head turn and tilt positions was remarkably high, except for persons wearing eye glasses or having a beard (for which it was zero). Correct results were obtained 91% of the time, for example, for full faces with no glasses or beard, and nearly 80% for a turned face with no glasses. The procedures are offered as an approach to computer classification of human faces and are analgous to those that might be used to discern to which of a class of objects (e.g., which model of tank) a potential target may belong.

Kuan (1984) explores how terrain map knowledge should be represented for spatial planning or route selection purposes by an autonomous robotic vehicle. He presumes that the DMA (Defense Mapping Agency) digital terrain data base (with roughly 10M X 10M resolution) would be used, since it contains both terrain elevation as well as cultural information. Cultural features include not only notation of the surface material (water, soil, trees) but also symbolic representation of roads, bridges, towers, etc. Details of such things as road width, whether or not paved, and one or two-way, are also stored. Another source of functional map information is the topographic map data base, such as represented by the U.S. Geological Survey Map. Sources such as these would provide obstacle information (e.g., big mountains, lakes, deep rivers) which would be combined with intelligence information regarding enemy positions, location of mine fields, and threat regions to provide delineation of areas that should not be penetrated. Spatial planning would then be performed on the basis of reconnaissance mission orders that are translated into operational requirements that serve as constraints or criteria for route planning (e.g., "Investigate the region between Mountain-1, Mountain-2, River-1, Bridge-1, and Bridge-2. Don't go to already occupied enemy regions; check on the valley region to be sure it is clear of enemy occupation; make sure that enemy is not traveling through River-1 by boats; and go back to the starting point"). Depending on the nature of the area being traversed, it may be better to use sensor systems to, for example, follow the road, rather than following specified grid positions as would be more reasonable in an open terrain situation. A "connectivity graph" may be used to determine how to travel through open terrain at minimum cost and is constructed by identifying the types of links possible between various points in the area (as determined by roads, bridges, or the lack thereof). At this point, the route planner can provide a more detailed route description, together with mobility factors, that will determine vehicle speed, the sensor required, and the algorithms to be used for processing sensor data. The route planner can also develop defensive/evasive action plans during the period it is monitoring route execution. For example, if enemy forces are encountered when the system is traveling through open terrain, the planner can find the closest high concealment region and plan how best to escape.

Gilmore and Semeco (1985) describe development work on a knowledge-based route planning system that could be used by autonomous ground vehicles for navigating through terrain. The system (known as TREK): "1) analyzes a digital map to generate a global route, 2) performs scene interpretation to generate local routes, 3) maintains a track of its position through scene matching" (i.e., by comparing the present scene to its stored digital map), and "4) uses knowledge-based processing to validate and improve preliminary plans in light of predetermined mission goals." The system is written in Lisp and runs on a Symbolics 3600 computer. From the discussion offered by the authors, it appears that the system is in an early stage of development, though they do plan to incorporate the system into two functional mobile robots (which process additional sensory inputs) that were donated by IBM for research into materials handling and manufacturing applications.

Conclusion

It is obvious that the problem of understanding how we understand image content is a complex one, even for biological vision and processing systems. As researchers attempt to model the necessary and sufficient characteristics in computer vision systems, the complexities of the problem become further appreciated.

This section can provide only a sampling of the voluminous literature in this area (over 25 books have been dedicated to the design of computer vision systems alone, according to Fu & Rosenfeld, 1984). The presentation has been an overview of some of the problems and current capabilities in image understanding. The reports by Havens and Mackworth (1983), Brazakovic and Tou (1984), and Connell and Brady (1985) should be consulted for more detailed treatments of the topic.

Bibliography

Bajcsy, R., Joshi, A., Krotkov, E. and Zwarico, A. (1985). LandScan: A natural language and computer vision system for analyzing aerial images. In Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Vol 2, (pp 919-921), Morgan Kaufmann Publishers, Inc., Los Altos, CA 94022.

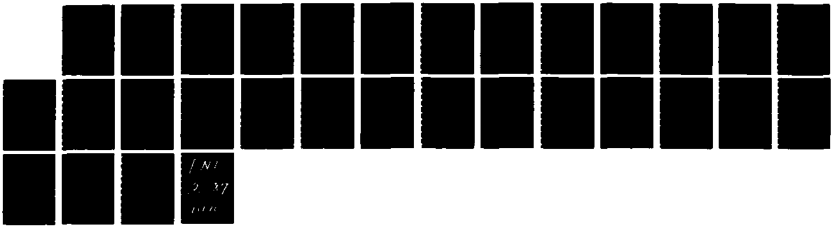
Barrow, H.G. and Tenenbaum (1982). Interpreting line-drawings as 3-dimensional surfaces. In J.E. Hayes, D. Michie and Y.H. Pao (Eds.), Machine Intelligence, Vol 10 (pp 227-238). Halsted Press: John Wiley & Sons, New York, NY.

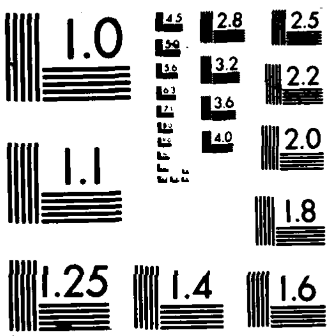
Brazakovic, D. and Tou, J.T. (1984). Image understanding via texture analysis. In Proceedings of First Conference on Artificial Intelligence, (pp 585-590). IEEE Computer Society Press, Silver Spring, MD 20910.

Cohen, P.R. and Feigenbaum, E.A. (Eds) (1982). The Handbook of Artificial Intelligence, Vol III. Heuristech Press, Stanford, CA.

AD-A175 456 AN ASSESSMENT OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TECHNOLOGY FO (U) HARRY G ARMSTRONG AEROSPACE MEDICAL RESEARCH LAB WRIGHT-PATTE W L MARTIN SEP 86 2/2

UNCLASSIFIED AAMRL-TR-86-040 F/G 6/4 NL





ROCOPY RESOLUTION TEST CHART

Connell, J.H. and Brady, M. (1985). Learning shape descriptions. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Vol 2, (pp 922-925), Morgan Kaufman Publishers, Inc., Los Altos, CA 94022.

Fu, K.S. and Rosenfeld, A. (1984). Pattern recognition and computer vision. *Computer*, *17*, 274-282, Oct.

Gilmore, J.F. and Semeco, A.C. (1985). Terrain navigation through knowledge-based route planning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Vol 2, (pp 1086-1088), Morgan Kaufmann Publishers, Inc., Los Altos, CA 94022.

Havens, W. and Mackworth, A. (1983). Representing knowledge of the visual world. *IEEE Computer*, *16*, 90-96, Oct.

Horn, P.K. (1980). Derivation of invariant scene characteristics from images. National Computer Conference Proceedings, Vol 49. (Reprinted in O. Firschein (Ed.), *Artificial Intelligence*, Vol IV, 1984, (pp 195-203), AFIPS Press, Reston, VA.)

Hubel, D.H. and Wiesel, T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology* (London), *160*, 106-154.

Hubel, D.H. and Wiesel, T.N. (1979). Brain mechanisms of vision. *Scientific American*, *82*, 84-97.

Kuan, D.T. (1984). Terrain map knowledge representation for spatial planning. In *Proceedings of First Conference on Artificial Intelligence*, (pp 578-584), IEEE Computer Society Press, Silver Spring, MD 20910.

Marr, D. (1982). *Vision*. W.H. Freeman, San Francisco, CA.

Mudd, G.R. (1980). Image understanding architectures. National Computer Conference Proceedings, Vol 49. (Reprinted in O. Firschein (Ed.), Artificial Intelligence, Vol IV, 1984, (pp 239-252), AFIPS Press, Reston, VA.)

Sakai, T. Nagai, M., and Kanade (1972). Computer analysis and classification of photographs of human faces. USA/Japan Artificial Intelligence Proceedings. (Reprinted in O. Firschein (Ed.), Artificial Intelligence, Vol IV, 1984, (pp 219-226), AFIPS Press, Reston, VA.)

Tenenbaum, J.M. Barrow, H.G., Bolles, R.C., Fischler, M.A., and Wolf, H.C. (1980). Map-guided interpretation of remotely-sensed imagery. National Computer Conference Proceedings, Vol 49. (Reprinted in O. Firschein (Ed.), Artificial Intelligence, Vol IV, 1984, (pp 201-218), AFIPS Press, Reston, VA.)

Winston, P.H. (1984). Artificial Intelligence. Addison-Wesley Publishing Company, Reading, MA.

CHAPTER 6

Natural Language Processing/Understanding

Introduction

Although speech recognition and speech synthesis hardware has been on the market for several years and has found application in everything from home computer games to control and warning systems for automobiles, the real challenge in this area has and will continue to be recognition of continuous, unrestricted speech. Since use of a constrained vocabulary of brief utterances may provide the pilot a valuable mode of interaction with his aircraft systems (including expert systems), this section is provided to indicate the technology available, techniques being explored to advance the technology, and some directions being taken to explore what would be required of a natural language system.

Speech Recognition and Coding

Doddington and Schalk (1981) describe how computers recognize speech. A most important characteristic is the formant frequency (i.e., the frequency at which the voiced energy peaks) which is due to the acoustic resonances of the mouth cavity as controlled by the tongue, jaw, and lips. Sundberg (1977) gives a particularly good account of how the formants and other sounds are vocalized. The first step in speech recognition is to transform the input signal into a set of features or parameters. The features may be, for example, the spectrum amplitudes of each of a set of perhaps 16 to 30 bandpass filters. Or, features may be defined to be the rate of zero-crossings (i.e. the number of times the voltage representation of speech changes algebraic sign, from plus to minus, or minus to plus) in each of several broadband frequencies to obtain an estimate of the formant count in each frequency band. In another technique, called linear predictive coding (LPC), the speech signal is represented by the parameters of a filter that best relate to the input speech. Regardless of the features extracted, they are typically averaged over a 10 to 20 millisecond interval and sampled 50 to 100 times each second. The most difficult part of the recognition problem is how to synchronize cr time-align the input with the reference patterns. Typically, alignment is performed at the beginning and ending of a word and features are identified within each of usually 16 equally spaced slices of time. This input pattern is then compared with the reference vocabulary words and a measure of similarity computed for each. The reference word producing the least difference becomes the "recognized" word. When large differences occur between the input pattern and all of the reference words, the recognizer rejects the input.

White (1978) provides an overview of several speech recognition techniques based on template matching. To automatically recognize speech, the problems of how to represent speech as compactly as possible (to conserve memory) and then how to search for template matching efficiently (to minimize processing time) must be solved. One means consists of converting the input speech to a series of N dimensional vectors (the signal space), which are then compared to stored reference utterances (templates). A single word may be decomposed into a number of vectors, based upon the occurrence of stop consonants or other definable boundaries between sounds. Dynamic programming may be used to stretch, or compress an unknown utterance to produce a better temporal alignment with template utterances.

Several techniques have been developed to represent speech as compactly as possible. In one class of recognizers, the auditory spectrum is divided into 16 to 29 frequency channels (channel bandwidths are proportional to their center frequencies) and a tabulation of log amplitude outputs of each channel is used to characterize the utterance. (Only the difference between adjacent channel outputs may be actually coded. This is typically referred to as a pulse code modulation (PCM) process and is a common technique to reduce the number of bits required to represent a complex signal). Data rates can also be reduced by eliminating steady state speech segments and periods of silence. A run-length coefficient is used to tell the synthesizer, for example, how many times to repeat an interval. (This is referred to as variable frame-rate encoding.) Another data reduction technique uses a principal components analysis approach to extract the principal dimensions or factors from the input data. White states that some of the more successful recognition devices represent speech as a sequence of events that may be assigned binary values. Although there may be more features than dimensions in the original signal, the simplicity of the binary comparison process apparently overcomes the complexity.

Two basic techniques are used to perform a dynamic search process. The first, the sequential pruning test, makes a left-to-right comparison between template and encoded utterance and eliminates templates early on, as they appear to not match. The second approach uses a hypothesize and test paradigm in which a compressed template data base is compared with the basis vectors representing speech. A comparison with the set of full templates hypothesized to match best is then made. A typical utterance can be reduced dramatically in the original compressed template comparison process without loss of accuracy. Finite state models may be evolved to represent state transition probabilities to provide further compressing capabilities in future systems.

Andrews (1984) discusses the fact that linear predictive coding (LPC) has associated with it a standard (called LPC-10) for government/military applications so that equipments across vendors can be made compatible. The major advantage of LPC is that it potentially allows coding of speech at much lower bit rates than popular alternatives (e.g., 1.2 K bps may be possible, as compared to 32 K or 64 K bps for PCM systems, for example) while maintaining high speech quality. The basic approach using LPC is to model the spectral shaping (performed by the glottis and vocal/nasal cavity of the human vocal tract) using a recursive, time-varying filter having time-varying coefficients. Once these coefficients are obtained, the driving function can be derived and corresponds to glottal excitation and vocal-cord vibration (for voiced sounds) and turbulent excitation (for unvoiced sounds). Both excitation level and rate of attack are encoded as well. LPC may also be used to synthesize speech by using a time-varying linear filter with predictive coefficients and white noise and/or a pulse train as the excitation signal. Andrews states that although LPC synthesis does not yet work well for the speech of women and children, improvements are being made (through "amplitude spectral shaping, adding random-phase spectral components in the voiced excitation signal, and randomly phased spikes in the conventional unvoiced excitation signal").

Levinson and Liberman (1981) provide a detailed discussion and analysis of the acoustic effect of the shape of the vocal tract on the resulting sound. The shape of the vocal tract is modeled as a transfer function representing the series of resonating cavities that modulate the original fundamental frequency (emitted by the vocal cords) in particular ways characteristic of particular sounds.

In terms of the supporting hardware, Andrews (1984) claims that VLSI technology is the most important recent development in speech coding equipment and projects that there will be rapid and dramatic cost and performance improvements in the near future. LPC voice coders are being especially impacted through the development of digital signal processing (DSP) chips which are much easier to program and much cheaper than their predecessors. The future for continuous speech recognition technology, Andrews predicts, is not nearly as bright, and the ability to recognize unrestricted continuous speech from any speaker remains a distant goal and will not be realized until AI progresses much further. Much of the problem has to do with the spelling and pronunciation ambiguity within the English language. In this regard, the Japanese have a distinct advantage with their language and have announced their intentions to have a speaker-independent, continuous and unrestricted speech recognition system by 1990.

Most systems commercially available in this country are limited to recognition of well-defined utterances of relatively small vocabularies (perhaps 200 words) by a single speaker. Continuous speech recognizers are available but suffer from very limited vocabularies (perhaps less than 50 dissimilar words). To make these systems speaker-independent forces yet another reduction in vocabulary size, since each member, or at least representative members, of the user population must provide his/her own template.

Averbuch et al. (1986) report on the laboratory development of an IBM PC based (and housed) speech recognition system that is able to recognize up to 5,000 words spoken singly. It is reported that the system (called Tangora, after the Guiness Book of World Records' fastest typist - 147 wpm for one hour) can be trained for the voices of individual speakers by using a 20-minute speech sample. The 5000 word vocabulary accounts for 92.5 percent of the most often used words out of a collection of 27 million words contained in a sample of busines letters and office memoranda. The modular architecture of the system permits expansion to a 20,000 word vocabulary (which would account for 97.6 percent of words in the sample) through the addition of one PC Expansion Unit. The processor works with a 12-bit A/D converted signal which it converts each one-hundredth of a second to a 20-dimensional vector (based on an auditory modeling procedure) which is then quantized and classified as one of 200 possible sound types. A decoder, having internal feedback from a hypothesis search section to a match engine section, operates on the input stream using three parallel paths so that each path handles one third of the total vocabulary. The data are deciphered by performing an acoustic match between the input stream and the statistics of its language model. Averbuch et al. do not provide particulars concerning error rate or recognition rate (although they do say the system operates in real time) but indicate that they are presently studying: 1) how well the Tangora system performs over a several-month period of time across a large sample of speakers; 2) the effects of ambient noise on its performance; and 3) several human factors issues related to its utility.

Cockpit applications of voice recognition systems suffer from a unique set of problems. Background noise can be a problem due to differences between what was present when the system was trained and when it is used in the combat environment. Similarly, differences in physical and emotional stressors present during training versus airborne and/or combat maneuvers present special challenges.

Understanding Natural Language

A natural language interface, whether it be through a verbal or a keyboard exchange, involves the generation of an idea by the user and a user translation of that idea into a statement that can be understood by the program. If all works as planned, the program will understand the statement and take appropriate action that produces some result. That result must, in turn, be translated to a statement the user can understand. Rich (1984) addresses the problems and design considerations involved in construction of the first half of this language understanding process, understanding of user utterances by the program. The major problem confronting the developers of English natural language understanding systems has to do with the ambiguities of the language (at both the word and sentence level) and the difficulty in accounting for all the different ways a single request for action might be phrased in a truely unrestricted system. To lessen the problem, one may restrict the number of words, word order, or syntax allowed. However, doing so defeats the purpose of using natural language to start with (i.e., minimizing the time required to learn the interface language). To translate from a natural language statement to something the program can understand typically involves three steps. First, the statement must be divided into its components (words) and those words matched to the list of words (dictionary/lexicon) the program understands, along with the context in which the words appear, and sometimes even a list of what sentence forms are acceptable. Often the ambiguity problem goes away when the context in which the word appears is considered. The next step in the understanding process involves a syntactic analysis (or parsing*) which may be conducted using either a top-down, or a bottom-up approach. Both approaches operate on the basis that a sentence must be constructed using a set of rules. For example, it must have a noun phrase, followed by a verb phrase. The noun phrase may contain a series of adjectives prior to the noun, etc. Such rules comprise the grammar of the language. The top-down approach is characterized by the diagramming method taught in high school. In the bottom-up approach, intermediate word groups are constructed in a "building-block" approach until the top constituent is included, at which time the parse is complete.

The last step required to understand the user's statement is called semantic processing. This step is often conceptually split into a determination of what the statement means, and a determination of what to do about it. To actually conduct a dialog, statements (especially incomplete ones) must be interpreted within the context of the preceding exchanges. For example, a reference to "he" must be understood as being the person just mentioned. Since natural language poses interpretation problems that artificial languages (such as programming languages) do not share, AI techniques must be applied to make the natural language interface viable.

Waldrop (1984) further addresses the problems involved in what it means to understand natural language. The fundamental problem for a language translation machine (say from Russian to English, for example) is that it should "know" a great deal about the world prior to working with its dictionary. It really needs a universal encyclopedia. The famous ELIZA program (Weizenbaum, 1966) which imitates a nondirective (Rogerian) psychotherapist, represents the other end of the spectrum, having only a pure stimulus-response capability without regard to the

* More specifically, parsing refers to the process of deciphering how words in a sentence are related to each other. Diagramming a sentence is one common approach.

context in which the utterance is made. According to Waldrop (1984) a step up from ELIZA was provided in the Ph.D. dissertation by Terry Winograd at MIT. His program (called SHRDLU) converts words into program fragments which are then ordered based on sentence structure. The program works with a simulated robot arm that manipulates simulated blocks on a simulated tabletop. To develop capabilities at higher levels requires not only machine recognition that different sentences can mean the same thing (e.g., "Bill bought the car from Fred" equals "Fred sold the car to Bill."), but also that the context in which the word appears will change the meaning (e.g., "The duck is ready to eat", or "Sue made the bed" as opposed to "Sue made an A on the test.").

Noam Chomsky, a linguist at MIT, has developed theories of "transformational" grammar. His theories provide quasimathematical rules for manipulating factors such as tenses, word order, and word endings. According to Waldrop, however, such formalism has proven to be too cumbersome to be used in practical computer programs.

Another approach to trying to understand the deep structure of language has been taken by Roger Schank at Yale University. His "conceptual dependency" model maps words into so-called "primitives" so that, for example, verbs that involve changing the physical location of an object (e.g., walk, move, lift) are mapped into a single primitive "ACT" called "PTRANS" (for physical transport). Similarly, "an attribute such as Mary is dead maps into a primitive STATE, Mary Health (-10)."* Rumelhart and Ortony (1977) describe similar primitive-based systems.

In summary, Waldrop cites Barbara Grosz of SRI International as stressing the large amount of inference and mutual accommodation that occurs below the level of literal word meaning. For example, the

* That is, since Mary is dead, her health is rated -10 on a scale from +10 to -10.

sentence "Can you pass the salt?" is not a request for a yes-no answer, but rather an attempt to elicit a particular response. A listener (as well as a computer) must understand what response is appropriate. At any rate, a great amount of work remains to be done before we understand what "understanding" really is.

Reichman-Adar (1984) carries some of the above ideas (by Rich, 1984) further by concentrating on how present utterances are related to previous ones in natural conversation. In a dialog, necessary constraints are placed on the participants in order that they not be confused as to each other's meaning. These constraints were proposed by Grice (1975) as a set of conversational maxims. Whenever any of these are violated, an "inappropriate" conversational move results. Examples of Grice's maxims are: a) The conversational contribution should be only as informative as required, not more nor less; b) The speaker must be relevant, while taking into account the fact that there are "different kinds and foci of relevance" and that it is alright to change the subject of the conversation; and c) The speaker must avoid obscurity and ambiguity in his/her expressions, and present ideas in an orderly fashion. Reichman-Adar uses these rules, together with extensive study of natural dialogs to form the basis of an abstract computational system for discourse processing that could be used to structure one module of a working computerized discourse system. The major elements of the system include cues, expectations, and segmentation, which provide capabilities corresponding to what occurs in natural conversation when topics are developed, suspended and resumed, without having to explain or comment at the junctures. At any point in time, the listener must be able to determine if what is being said is done so to illustrate, support, or disagree with previous statements, or if it represents a shift to a previous topic or to a new topic altogether. To be useful as a module in a computerized discourse system, sufficient computational capability must be present so the program would not be confused and have to ask, "But what does this have to do with what we were talking about?", any more often than a human listener might. The program must maintain a discourse frame of reference. That is, it must keep track of what is currently being said and decide how it relates to previous discourse. Similarly, the program must be able to determine whether the current talk is an embellishment/continuation of previous utterances, or is the start of a new topic having a different frame of reference. Such conversational moves may present a claim, give support to a claim, explain a claim, challenge a claim, shift to a new topic, or resume an earlier exchange. The current discourse frame of reference also must be considered in the context of the previous frames of reference that have occurred. This is analogous to figure-ground relationships found in other sensory processes (re: Gestalt psychology) and provides strong organizational qualities to our perceived world. The current frame of reference provides the "figure," while the previous discourse serves as the "ground." Conversational moves establish expectations as to what moves might follow. For example, a series of utterances followed by the clue words, "But, first" tells the listener (and should tell the program) to be ready to shift gears into either a new topic, or to a piece of information he/she will need in order to understand some future segment of discourse. Much of the knowledge shared by listeners and speakers has to do with the components of allowable conversational moves. For example, a bit of technical discourse will be quite different when talking to a grade school child, as opposed to one's colleague, much of the difference owing to the nature of the conversational moves required.

The discourse module Reichman-Adar proposes keeps track of all the relationships discussed above, together with interactions among them in dedicated registers. As examples, the "Expectation-List register" consists of a list of discourse expectations that are as yet unfulfilled; the "Domain-Constraints register" is composed of a list of the points previously conceded by the conversants in an argument to assure that these are not used again; and the "speaker register" maintains track of whose conversational move is presently being processed. Although the Reichman-Adar approach to machine processing of natural discourse must address the difficult subtleties of the English linguistic system, these must be treated if the flavor and content of our meta-communication processes are to be translated into a form useful for unrestricted person-machine dialog.

Sedelow (1976) provides an analysis of referential linkages in extended strings of written words. The author argues for the application of a general-purpose thesaurus for determining semantic frames of reference and semantic differences among various elements of the word string. An extensive analysis of the meaning(s) of a short 6-line passage was undertaken to illustrate the principles that would have to be exercised, using the thesaurus along with a dictionary and parser. The primary technique would be to use the dictionary, parser, and thesaurus in combination to evaluate the semantic distances between the alternative word meanings in order to arrive at a best possible interpretation of the passage. Sedelow feels the importance of computer-based discourse analysis for natural language applications cannot be overemphasized.

Some Applications Research

Doddington and Schalk (1981) point out the problems with which speech recognition systems have to deal. The primary problem is that a particular speech sound is voiced according to what sounds come before and after it. For example the "t" and "r" sounds in the sequences "what time" and "chair ramp" will be coarticulated as single sounds, respectively. These and other mergings and blendings at the boundaries between words in the flow of natural language cause word end points to be very difficult to determine reliably and are the source of almost all word-recognition errors by discrete-word recognizers. Connected-speech recognizers do not suffer from this problem to such an extent and are therefore able to perform better in discrete speech applications as well. Recognition of continuous speech (for short word sequences using a small vocabulary) requires that the speech signal be divided into intervals corresponding to particular acoustic patterns that may correspond to words, syllables or phonemes. The more pronounced the pause between words, the better these systems perform.

Levinson and Liberman (1981) state that one way to reduce the complexity of the continuous speech recognition problem is to take advantage of the allowable sequences of words in a sentence or syllables in a word and define, within the computer, every possible transition between states (word-to-word or syllable-to-syllable) along with their associated probabilities. Based on the acoustic measurements, the recognizer selects the path corresponding to the product of transition probabilities that is found to be the largest.

According to Doddington and Schalk (1981) the second most common problem concerns the inconsistencies of word pronunciation by a single speaker over time. Factors that contribute to this involve anything in the environment that may affect noise and reverberation, microphone placement, as well as the loudness of speech. Recognition performance can deteriorate significantly under changes as small as $\pm$ 6 dB in the input level, according to these authors.

Doddington and Schalk evaluated a total of seven speech recognizers that ranged in price from $500 to $65,000 and that represented speaker-dependent and speaker-independent, as well as connected and discrete-word recognition systems. The test trials consisted of a 20-word vocabularly composed of the 10 digits (zero through nine) and 10 command words (i.e., yes, no, start, stop, go, help, rubout, erase, enter, and repeat). After an initial enrollment session, consisting of ten passes through the vocabulary, eight test sessions (all using discrete-word inputs) were conducted for each of eight men and eight women over nearly a two-month period. Their data indicate that substitution errors (mistaking one word for another) spanned the range from .2 percent (for one of two $65,000 systems tested) to 12.6 percent (for the $500 system*). In all but one of the seven systems tested, error rates were higher for women than for men (by an average of 1.6 percent). Doddington and Schalk insist that due to differences in the speaker population, background noise, microphone characteristics and vocabulary used, the only reliable way to estimate performance for some particular set of conditions is to actually measure performance under those conditions.

The design of knowledge-based help systems has been addressed by Fischer, Lemke, and Schwab (1985). Although their work has been performed in the context of keyboard text editing systems, their ideas and methods are applicable to verbal natural language dialogs as well. Knowledge-based help systems should be based on a model of the task to be performed, together with a model of the user and his/her progression through the elements of the task. Fischer et al. describe two operational prototype systems, representing both passive and active capabilities. Passive help systems may supply keywords or synonym lists to provide assistance. Active help systems may take the form of canned error messages, but must include a model of the task and user to recognize suboptimal (not just erroneous) user actions so that appropriate messages can be formulated and presented. This would require that a metric of the adequacy of user actions be developed and these authors point out that although this is a very difficult task, making the metric visible to the user would allow him/her to change it if, for example, the help system forces the user to do something in a way that was really not desirable.

These authors describe their attempts to develop both the active and passive help systems. In order to better understand the real needs of the user, they asked an expert with their text editor system (called BISY) to assist persons of various proficiencies in its use and recorded

* The manufacturer of this system has since used the data from this study to revamp the algorithm used so as to reduce substitution errors to an advertised 6.4 percent level.

the types of help requested. Based on this information, a help system (PASSIVIST) was developed which allows the user to type out the question he/she has. PASSIVIST then picks out the key words in the question and indicates to the user what it has understood. For example, the question "How can I delete the next line?" is responded to by denoting the key words to be "delete next line," together with an indication that the words "How, can, I, the" were ignored. This has the obvious advantage of allowing misconceptions to be corrected and provides the basis for development of a model by and for each side of the human-computer interface. Once the user is satisfied PASSIVIST is correct in its interpretation of his/her problem, a solution is provided by displaying the series of commands required to solve the problem. Another system, called ACTIVIST, was developed to similarly explore the characteristics required for an active help system that attends to suboptimal user behavior. ACTIVIST recognizes what the user is doing or wants to do, evaluates how the user tries to achieve his goal, constructs a model of the user based on the results of the evaluation, and decides when to interrupt and in what way. During (or at the conclusion of) an editing session, ACTIVIST shows the model of the user that has been built up, including how often optimal, suboptimal, or wrong commands were used, as well as how often messages were provided to the user. As the frequency of optimal commands of a particular variety increases, ACTIVIST will modify its tutorial strategies.

Fischer et al. conclude that as editing functions and user actions become more sophisticated, the number of actions that must be monitored will exceed the computational power available. The solution to the problem will be to restrict the number of inferred action sequences the program monitors, deleting those that have been done reliably in the past.

Biermann, Rodman, Rubin, and Heidlage (1985) describe an experimental evaluation of an interactive, natural language system that uses a commercially available discrete speech recognizer (Nippon

Electric Corp. Model DP-200) that requires a pause of approximately 300 ms between utterances. The purpose of their research was to determine: how fast subjects would learn to speak in a way that the machine would recognize their sentences; what word and sentence error rates would result; how fast commands could be spoken; and how acceptable users felt the system was. The context of the user interaction with the system was the solution of a set of three simultaneous linear equations in three unknowns. Although subjects (volunteers from a college mathematics course) were free to use any method to solve the problem, most addressed the problem in the traditional way by producing an identity matrix in the first three columns (which provides the solutions in the fourth column). Prior to the experimental sessions, each subject was asked to train the machine using a set of 100 words, each spoken with rising, flat, and falling inflection. Then the system was tested by having subjects speak commands typical of the type to be used during experimental sessions. The recognizer signalled a rejection by an auditory beep over a headset. A "recognized" word was displayed on the CRT. Subjects were asked to end each sentence with the word "over," which indicated a request to the system for command execution. Sentences could be aborted by the word "forgetit." The results indicated that about 77 percent of the over 6000 input sentences were correctly processed. The "forgetit" terminations accounted for another 13 percent of total transactions. The remaining failed transactions were due to user errors (3.15 percent - due to too many or too few words for the parser to handle), incorrect substitution errors (2.05 percent - due to the recognizer making an incorrect substitution which was not noticed by the subject), system errors (1.68 percent - computer did not respond correctly to a syntactically correct sentence), change of mind (1.25 percent - subject corrected himself), unimplemented commands (.66 percent - subject used an illegal word), and logout (.59 percent - subject terminated the process due to time-consuming parsing process). As to the question of learnability, these researchers found that about one hour of practice was sufficient for most subjects to learn the system, and they felt that a motivated user could certainly use the system in under two hours. Subjects averaged 46.5 words per minute, with a range of 40.8 to 57.0. Their transaction rate increased from a little over four per minute during the early sessions (first 1/6th) to over six per minute during the last block (last 1/6th) of the sessions. As for user acceptance, the subjects displayed no serious hostility toward the system and all completed the experiment "willingly and cooperatively."

Summary

A fine general discussion of the linguistic considerations needed to make computers understand natural language can be found in the book "Artificial Intelligence" by Patrick H. Winston 1984, (pp 291-334) and is recommended for those readers who may have particular interest in that area.

In addition, Lea (1980) provides a comprehensive review of speech recognition research and technology (up to about 1978) in his book "Trends in Speech Recognition". This source is highly recommended for those wishing to gain a thorough appreciation for virtually all aspects of the speech recognition problem, applications, advantages and disadvantages of automated speech recognition, together with research status and needs. The book is a compilation of 27 original papers authored by leading experts in this field.

It should be clear from the reports reviewed in this section that, in order to generate a program that would demonstrate artificial intelligence in terms of "understanding" an unrestricted natural language flow and to respond in an intelligent fashion, more artificial intelligence would have to be built into the program than we now know how to provide in any concrete way. Such English language systems are probably several decades away from realization. In the meantime, useful systems will be developed and fielded, which, though they lack a true natural language dialog capability, provide the trained user a mode of machine interaction that will be welcomed and, if designed properly, will even reduce workload.

Cockpit applications of these systems could be especially powerful for control of selected subsystem operations during periods of peak stress, if and only if the emotional and environmental stressors of the moment do not reduce the reliability of the voice recognition system below acceptable standards.

Bibliography

Andrews, H.L. (1984). Speech processing. Computer, 315-324, Oct.

Averbuch, A., Bahl, L. Balcis, R., Brown, P., Cole, A., Daggett, G., Das, S., Davis, K., DeGennaro, S., de Souza, P., Epstein, E., Fraleigh, D., Jelinek, F., Katz, S., Lewis, B., Mercer, R., Nadas, A., Nahamoo, D., Picheny, M., Shichman, G., and Spinelli, P. (1986). An IBM PC based large-vocabulary isolated-utterance speech recognizer. In Proceedings of International Conference on Acoustics, Speech and Signal Processing (pp 53-56), (IEEE CH2243-4/86/0000-0053, ICASSP 86, Tokyo, Japan).

Bierman, A.W., Rodman, R.D., Rubin, D.C., and Heidlage, J.F. (1985). Natural language with discrete speech as a mode for human-to-machine communication. Communications of the ACM, 28, (6), 628-636, Jun.

Doddington, G.R. and Schalk, T.B. (1981). Speech recognition: Turning theory to practice. IEEE Spectrum, 26-32, Sep.

Fischer, G., Lemke, A. and Schwab, T. (1983). Knowledge-based help systems. In Proceedings of Computer-Human Interface Conference, Association for Computing Machinery, (pp 161-167), Apr.

Grice, H.P. (1975). Logic and conversation. In P. Cole and J.Morgan (Eds.), Syntax and Semantics, Academic Press, New York, NY, (pp 41-58).

Lea, W.A. (Ed.). (1980). Trends in speech recognition. Prentice-Hall, Inc. Englewood Cliffs, NJ 07632.

Levinson, S.E. and Liberman, M.Y. (1981). Speech recognition by computer. Scientific American, 64-76, Apr.

Reichman-Adar, R. (1984). Extended person-machine interface. Artificial Intelligence, 22, 157-218.

Rich, E. (1984). Natural-language interfaces. Computer, 17, (9), 39-47, Sep.

Rumelhart, D.E. and Ortony, A. (1977). The representation of knowledge in memory. In R.C. Anderson, R.J. Spiro and W.E. Montague (Eds.) Schooling and the Acquisition of Knowledge. Lawrence Erlbaum Associates, Hillsdale, NJ.

Sedelow, S.V. (1976). Analysis of "natural" language discourse. National Computer Conference Proceedings, Vol 45. (Reprinted in O. Firschein (Ed.), Artificial Intelligence, Vol IV, 1984, (pp 171-178), AFIPS Press, Reston, VA.)

Sundberg, J. (1977). The acoustics of the singing voice. Scientific American, 82-91, Mar.

Waldrop, M.M. (1984). Natural language understanding. Science, 224, 372-374, Apr.

Weizenbaum, J. (1966). ELIZA - A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9, (1), 36-44, Jan.

White, G.M. (1978). An overview of several fundamental recognition techniques. USA/Japan Artificial Intelligence Proceedings. (Reprinted in O. Firschein (Ed.), Artificial Intelligence, Vol IV, 1984, (pp 179-183), AFIPS Press, Reston, VA.)

Winston, P.H. (1984). Artificial Intelligence. Addison-Wesley Publishing Co., Reading, MA.

CHAPTER 7

Summary Comments on the Development and Application of Artificial Intelligence and Expert Systems

An expert system can be expected to usually make poorer decisions than the experts who assisted in the development of the system, even though the expert system may arrive at a decision faster. The poorer expert system performance is expected due to the inability of the experts to consider all possible contingencies during the rule formulation phase of the knowledge acquisition process, or an inability to describe what they are actually doing in the decision-making process. In addition, when new rules are added to a knowledge base in order to upgrade an existing expert system, the complexity of determining if and how old rules must be modified (or deleted) to be compatible with the new rules is not minor and may result in logical inconsistencies that may be extraordinarily subtile, leading to erroneous (or unintended) decisions some proportion of the time.

On the other hand, the expert system may outperform the human decision maker, even an expert in the field, for a variety of reasons. If the expert system had been developed by a team of experts, the chances are great that their group decisions (that were incorporated into the logic structure of the expert system) are of higher quality than any single expert's might be (see Jewell and Reitz (1981) for a discussion of individual versus group decision-making). Since the expert system is deterministic in nature, it will not have the frailty of being variable in its output. This offers considerable advantage over the human, especially under emotional, environmental, or high workload conditions in which human thought processes may yield less than optimal decisions.

Similarly, with VLSI technology comes the promise of very rapid processing of a relatively large number of rules (comprising a comprehensive expert system) so that problems having significantly greater complexity than a single human could possibly deal with in a short period of time may be solved more rapidly and properly/accurately by an expert system. At a minimum, expert systems can be expected to significantly assist less experienced personnel in the more routine aspects of a process or task and play the role of advisor early in the learning process.

In concert with opinions voiced by business representatives, both in and outside of the aerospace industry, as well as by some academicians, it is apparent that initial expert system development efforts will be most fruitfully spent in the areas that are easiest to develop and implement. Decision aiding systems, diagnostic systems for aiding maintenance tasks, as well as mission, project, and route planning systems are among the most likely candidates for early implementation. Other strong contenders will be shells from expert systems developed within industry to solve routine production, process control, and/or maintenance problems. Classification of threat signals, sensor suite management, navigation system management, and automated software generation are among the next most likely candidate tasks for expert systems applications.

The projected dramatic reduction in symbolic processing hardware and software costs will continue to foster greater interest and activity in expert system developments throughout government, industry, academia, and the home market areas.

The greatest source of real and potential problems in the implementation of expert systems remains to be centered squarely at the user interface. The most prominent of these concern the presumed experience level of the user (by the system designer), the construction and maintenance of a model of the user by the system, the ease with which the user can develop and maintain an accurate model of the system and what it is doing, and the extent to which the user is satisfied by the response (quantitative and qualitative) of the system. Evaluation of command, menu and iconic interfaces has shown that no particular type emerges as inherently better than the others and that a good example of any one of the three types will be judged to be good by novice and expert users alike, thus dispelling the notion that there is a learnability versus usability tradeoff for these interfaces.

Another rather surprising result that bodes well for the use of constrained vocabularies in problem solving situations (in the cockpit) is that self-limited brevity actually seems to augment the problem solving process, indicating that communication under these conditions may actually be enhanced.

Of all the challenges to the AI community, the automated understanding of image content stands as the most difficult from all perspectives. To operate in real-time, or even near-real-time requires a level of sophistication of processing hardware, software and supporting algorithms/models that will likely not be achieved in any comprehensive sense for several decades. One of the most difficult problems in machine recognition (that the human performs easily) is identification of the content (the background or ground) in which the object of interest (the figure) is situated. Most often, the human will recognize the object, based on what he/she knows about the context of the situation. The storage, piecing together, and application of this contextual information by the computer will constitute a major step in AI capabilities for airborne, industrial, medical, and other applications as well.

Running a close second to the difficulty of the image understanding problem is the problem of recognizing unconstrained, continuous speech. Machine recognition and understanding of the context in which the talk is situated poses many of the same types of problems for continuous speech as it does for machine visual systems. Similarly, partially blended (or nonexistent) boundaries between objects, whether they be words in a continuous stream, or elements in a scene, generate similar requirements on the part of the receiver/sensor to rely on contextual cues to extract meaning and understanding.

Ambiguities in the English language provide another sticky problem for discrete as well as continuous speech recognizers. Another major problem has to do with the variety of ways a single statement, request for action, or question may be phrased and the associated equivalences that need to be provided for in the program.

Superimposed on all these difficulties is the fact that even a single speaker will not utter the same word or sentence exactly the same each time and this variability is added to whatever environmental differences (e.g., noise, reverberation) may also be present during various occurrences of an utterance.

In the face of all the expressed concerns with machine understanding of natural language, it is nevertheless reasonable to expect that a trained user with a high quality, *trained system* (using a constricted vocabulary) may find this mode of man-machine interaction highly desirable and even necessary if the user is to perform at highest efficiency.

In conclusion, when considering the development and application of AI and expert systems in general, we must recognize that in order to maximize our probability of success, we must understand the science of the problem being addressed. An example of what happens when this is disregarded was mentioned in Chapter 3, in connection with the artificial heart program. That effort failed primarily because no one understood the rejection mechanisms of the human body, yet political motivation caused the program to push the limits of its technology beyond the breaking point. An analogous situation could easily occur in the AI and expert systems area. The great caution that must be levied is that we not try to extend the capabilities of these programs beyond the limits justifiable by the knowledge base that is available to support them.

By now there has been a sufficient body of experience accumulated by developers and appliers of expert systems to instill, in some cases, modest reservation, and in others more extreme views as to the pitfalls awaiting the purveyors of this technology. When programs are developed and applied by users blindly, without questioning the assumptions or logical flow underlying suggested courses of action output by these programs, totally unpredictable problems can (and most certainly will) arise that cause the credibility of the programs to suffer to the extent that further development and use of these tools will likely not be pursued.

On the other hand, given proper support to the developers of these systems, and realistic expectations by users as to what is achievable with the programs, real and significant advancements in the sophistication, utility and application of AI and expert systems can and will continue to be made.

Bibliography

Jewell, L.N. and Reitz, H.J. (1981). Group Effectiveness in Organizations. Scott, Foresman, Glenview, IL.

Glossary of Abbreviations, Acronyms, and Special Terms

AAMRL - Harry G. Armstrong Aerospace Medical Research Laboratory
A/D - analog-to-digital (conversion)
AFIT - Air Force Institute of Technology
AFWAL - Air Force Wright Aeronautical Laboratories
AI - artificial intelligence
ALERT - Algorithm, Learning, and Recognition Technique
ALV - autonomous land vehicle
ART - Automated Reasoning Tool
ATF - Advanced Tactical Fighter
ATR - airborne transportable rack
AWACS - Airborne Warning and Control System
BBN - Bolt Beranek and Newman (Inc.)
BETA - Battlefield Exploitation and Target Acquisition
BIT - built-in-test
bps - bits per second
BRASS - Battlefield Robotic Ammunition Service System
CCD - charge coupled device
CEP - circular error probability
C^3I - command, control, communication, and intelligence
DARPA - Defense Advanced Research Projects Agency
DMA - Defense Mapping Agency
DOD - Department of Defense
DSP - digital signal processing
DTIC - Defense Technical Information Center
ECM - electronic countermeasure
EPES - Emergency Procedures Expert System
ETS - Expertise Transfer System
EW - electronic warfare
FDS - Flight Design System
FIXER - Fault Isolation Expert for Enhanced Reliability
FLIR - forward looking infrared

GPS - Global Positioning System

IMAPS - Interactive Mission Analysis Planning Station

INS - Inertial Navigation System

KEE - Knowledge Engineering Environment

KES - Knowledge Engineering System

Lisp - list processing (language)

LPC - linear predictive coding

MCAIR - McDonnell Aircraft Company

MIPS - millions of instructions per second

MOS - metal-oxide-semiconductor

ms - milliseconds

MTBF - mean time between failures

NAVEX - Navigation Expert System

PC - personal computer

PERT - Program Evaluation and Review Technique

PCM - pulse code modulation

RENEX - Rendezvous Expert System

SDI - Stragetic Defense Initiative

SITAN - Sandia Inertial Terrain

TAC - Tactical Air Command

TI - Texas Instruments

TTR - time to repair

VAX - Digital Equipment Corporation's name for a family of their computers

VHSI - very high speed integrated (circuit)

VLSI - very large scale integrated (circuit)

* U.S. GOVERNMENT PRINTING OFFICE 1986 - 748-001/40218

6c. ADDRESS (*City, State, and ZIP Code*)		7b ADDRESS (*City, State, and ZIP Code*)			
Wright-Patterson AFB OH 45433-6573					
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.
		62202F	7184	11	45

11 TITLE (*Include Security Classification*) AN ASSESSMENT OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS TECHNOLOGY FOR APPLICATION TO THE MANAGEMENT OF COCKPIT SYSTEMS (U)

12 PERSONAL AUTHOR(S)
Martin, Wayne L.

13a. TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
Technical	FROM ________ TO ________	1986 September	126

16. SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	Artificial Intelligence
			Expert Systems Technology
			Cockpit Systems

END

2-87

DTIC