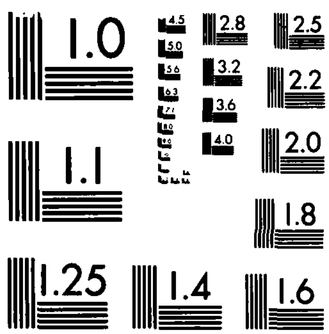
AD-A161 139 AUTOMATED INFORMATION MANAGEMENT TECHNOLOGY (AIM-TCCH): 1/3
CONSIDERATIONS FO (U) MACAULAY-BROWN INC FAIRBORN OH
W REITMAN ET AL MAY 85 AFAMRL-TR-85-042

UNCLASSIFIED F33615-82-C-0513 F/G 9/2 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

AFAMRL-TR-85-042

AD-A161 139

# AUTOMATED INFORMATION MANAGEMENT TECHNOLOGY (AIM-TECH): CONSIDERATIONS FOR A TECHNOLOGY INVESTMENT STRATEGY


*WALTER REITMAN*
*RALPH M. WEISCHEDEL*
*BOLT BERANEK AND NEWMAN INC.*
*KENNETH R. BOFF*
*AIR FORCE AEROSPACE MEDICAL RESEARCH LABORATORY*
*MARK E. JONES*
*MACAULAY-BROWN, INC.*
*JOSEPH P. MARTINO*
*UNIVERSITY OF DAYTON RESEARCH INSTITUTE*


*MAY 1985*

*Approved for public release; distribution is unlimited.*




*AIR FORCE AEROSPACE MEDICAL RESEARCH LABORATORY*
*AEROSPACE MEDICAL DIVISION*
*AIR FORCE SYSTEMS COMMAND*
*WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6573*


85 11 12 012

## NOTICES

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Please do not request copies of this report from Air Force Aerospace Medical Research Laboratory. Additional copies may be purchased from:

National Technical Information Service
5285 Port Royal Road
Springfield, Virginia 22161

Federal Government agencies and their contractors registered with Defense Technical Information Center should direct requests for copies of this report to:

Defense Technical Information Center
Cameron Station
Alexandria, Virginia 22314

## TECHNICAL REVIEW AND APPROVAL

AFAMRL-TR-85-042

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
Air Force Aerospace Medical Research Laboratory

AD-A161 139

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS |
| 2a. SECURITY CLASSIFICATION AUTHORITY | 3. DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited. |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S) | 5. MONITORING ORGANIZATION REPORT NUMBER(S)<br>AFAMRL-TR-85-042 |
| 6a. NAME OF PERFORMING ORGANIZATION<br>MacAulay-Brown, Inc.<br>6b. OFFICE SYMBOL (If applicable) | 7a. NAME OF MONITORING ORGANIZATION<br>AAMRL/HEA |
| 6c. ADDRESS (City, State and ZIP Code)<br>3989 Colonel Glenn Highway<br>Fairborn, OH 45324 | 7b. ADDRESS (City, State and ZIP Code)<br>Wright-Patterson AFB OH 45433-6573 |
| 8a. NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b. OFFICE SYMBOL (If applicable) | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER<br>F33615-82-C-0513 |
| 8c. ADDRESS (City, State and ZIP Code) | 10. SOURCE OF FUNDING NOS. |

| PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT NO. |
|---|---|---|---|
| 62202F | 7184 | 26 | 06 |

11. TITLE (Include Security Classification)
SEE REVERSE

12. PERSONAL AUTHOR(S)
SEE REVERSE

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Yr., Mo., Day) | 15. PAGE COUNT |
|---|---|---|---|
| Final | FROM ____ TO ____ | 1985 May | 218 |

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

| FIELD | GROUP | SUB. GR. |
|---|---|---|
| | | |
| | | |

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Artificial Intelligence; Design; Automated Information Management; Expert Systems; Command, Control, Communication; (Continued)

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

The Automated Information Management Technology Program (AIM-Tech) has focused on evaluating the key factors to an investment strategy for implementing artificial intelligence technology in three technical domains: systems design; pilot/aircrew automation; and command, control, and communications ($C^3$). A state-of-the-art review of artificial intelligence (AI), functional specifications for future AI aided systems, and required AI capabilities are discussed. The state-of-the-art review assesses eight technology areas: expert systems and knowledge engineering, natural language, knowledge representation, computer vision, tutoring and training, planning and problem solving under real world conditions, AI tools and environments, and speech. Functional specifications are classified into three groups for each technical domain: communications, expert understanding, and decision aiding. Futuristic scenarios were constructed in which information and control management choke points are resolved by the hypothetical application of machine intelligence for each of the three domains. The technology demands of each scenario are then
(Continued)

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☒ DTIC USERS ☐ | 21. ABSTRACT SECURITY CLASSIFICATION<br>UNCLASSIFIED |

| 22a. NAME OF RESPONSIBLE INDIVIDUAL | 22b. TELEPHONE NUMBER (Include Area Code) | 22c. OFFICE SYMBOL |
|---|---|---|
| Dr. Kenneth R. Boff | (513) 255-7596 | AAMRL/HEA |

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

11. Title

AUTOMATED INFORMATION MANAGEMENT TECHNOLOGY (AIMTECH): CONSIDERATIONS FOR A TECHNOLOGY INVESTMENT STRATEGY (U)

12. Personal Author(s)

Reitman, Walter
Weischedel, Ralph M.

Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, MA 02238

Boff, Kenneth R.

AAMRL/HEA
Wright-Patterson AFB OH 45433-6573

Jones, Mark E.

MacAulay-Brown, Inc.
3989 Colonel Glenn Highway
Fairborn, OH 45324

Martino, Joseph P.

University of Dayton Research Institute
Jesse Phillips Center
Dayton, OH 45469

18. Subject Terms (Cont'd)

Image Understanding
Information Management
Knowledge Engineering
Knowledge Representation
Natural Language
Pilot/Aircrew Automation
Planning
Problem Solving
Speech
Technology Investment Strategy
Training

19. Abstract (Cont'd)

analyzed and matched with technology milestones identified in each area and estimates of when the milestones might be achieved. The current and near-term AI state-of-art pertaining to the milestones and technology demands of the scenarios are summarized in terms of cost and levels of effort needed.

# SUMMARY

The objective of AIM-Tech is to produce the essential resources to enable development of an investment strategy of artificial intelligence (AI) technology for the Air Force Aerospace Medical Research Laboratory (AFAMRL).

The report is organized into five sections: (1) Introduction; (2) AI state-of-the-art review; (3) functional specifications for future AI aided systems; (4) required AI capabilities; and (5) the AIM-Tech investment strategy. The Introduction describes the AIM-Tech Program, its background and rationale. The project focused on three technical domains as areas for potential AI applications: (1) systems design; (2) pilot/aircrew automation; and (3) command, control, and communication. These technical domains were evaluated during a three day "brainstorming" workshop. Fifty professionals, both local and nationally based, participated representing the military, DoD contractors, and academia. The workshop product was distilled into a listing of information and control management choke points in each of the three domains.

Section two, "AI: State-of-the-Art Review," provided a detailed assessment of eight AI technology areas. These are:

- o expert systems and knowledge engineering
- o natural language
- o knowledge representation
- o computer vision (image understanding)
- o tutoring and training
- o planning and problem-solving under real world conditions
- o AI tools and environments
- o speech

Each area is reviewed separately with discussions of background (including glossary of terms), operational applications, techniques for effective operational systems, principal areas of research, approaches that failed, major laboratories with key contact points, and recommended key references for further reading.

Section three, "Functional Specifications for Future Systems," lists for each technical domain the desired elements or functional specifications which were generated at the workshop. The functional specifications are in three groups for each technical domain: communications, expert understanding, and decision aiding. Futuristic scenarios are constructed in which information and control management choke points are resolved by hypothetical application of machine intelligence in each of the three domains. The technology demands of each scenario are then analyzed and matched with technology milestones identified in each AI area with estimates of when the milestones might be achieved.

The capabilities are further discussed in section four, "Required Artificial Intelligence Capabilities." This section begins by presenting some background material on milestones and technological forecasting. The capabilities required by the scenarios are once more reviewed and segmented into each of the eight AI areas. In each AI area, the current and near-term state-of-the-art is reviewed along with the associated milestones and their projections. The milestone projections are summarized at the end of this section with respect to person-years and time needed for development.

Section five, the "Considerations for a Technology Investment Strategy," evaluates the summarized data from section four and provides guidance and caveats for using the AIM-Tech data as a management resource.

**ACKNOWLEDGEMENTS**

This work was supported by Air Force Aerospace Medical Research Laboratory (AFAMRL) Director's Funds (LDF 84-12) and funds from AFAMRL Human Engineering (HE) Division project 7184, task 26, work unit 06. The authors are grateful to Dr. George Mohr, Laboratory Commander, Mr. Charles Bates, HE Division Chief and to Dr. Thomas Furness III, Branch Chief for their support of this program. Initial conversations with Mr. John Sinacori stimulated the eventual organization of this program; his ideas and efforts are gratefully acknowledged. Ms. Gloria Calhoun, AFAMRL/HEA, made vital contributions to the success of this effort during its formulation and initializing phases. MacAulay-Brown, Inc., as prime contractor, Bolt Beranek and Newman (BBN), Resource Associates and the University of Dayton Research Institute supported various phases of this effort. Mr. Glen Gallaway and Mr. Mark Jones of MacAulay-Brown, Inc. made important contributions to managing this effort. In particular, Mr. Gian Cacioppo of MacAulay-Brown, Inc. was responsible for the professional level of management that the program has enjoyed. Dr. Sheldon Baron, Vice-President, Computer and Information Sciences Division, BBN, provided able management and much needed advice and assistance. Mr. Scott Sheely, President, Resource Associates and Dr. Tom Rueth, Eastway Corporation expertly organized and orchestrated the AIM-Tech Workshop Phase A of this program. We are indebted for the contributed efforts of the persons who served as moderators for the AIM-Tech Workshop. Ms. Judy Williams, MacAulay-Brown, Inc.; Dr. Janet Lincoln, New York University; Major Robert Eggleston, AFAMRL/HEA; Mr. Tim Anderson, AFAMRL/BBA; Mr. Joseph Wohl, Alphatech, Inc.; and Captain David Leupp, AFAMRL/HEC. This demanded much time and energy preparing for, participating in and summarizing the results of the working groups. Moreover, these individuals made important conceptual contributions to the way the workshop was conducted and the analysis of workshop products. This report is the result of much collaboration. In each of the AI subject areas, experts from BBN were called upon to help in drafting and/or editing text in their respective areas of expertise. These contributions include the following: N. Sridharan & A. Stevens: Expert Systems; R. Scha: Natural Language; J. Schmolze: Knowledge

Representation; R. Schudy & R. Bobrow: Computer Vision; A. Rubin & B. Bruce: Tutoring and Training; N. Sridharan: Planning; J. Gibbons & D. Allen: AI Tools and Environments; M. Krasner & J. Wolf: Speech. A project of this complexity could not have operated effectively without the caring and dedicated administrative assistance provided by Tanya Ellifritt, AFAMRL/HEA; Sonia Alexander, Tammy Hanser, Mary Jo Hawkey, Jeanie Lauber, and Colleen Morton of MacAulay-Brown, Inc.; Beverly Tobiason, Phyllis Pizzi and Mira Sussman of BBN.

## TABLE OF CONTENTS

**TABLE OF CONTENTS**
(Cont'd)

## TABLE OF CONTENTS
(Cont'd)

TABLE OF CONTENTS
(Cont'd)

## LIST OF ILLUSTRATIONS

## LIST OF TABLES

# Section 1. INTRODUCTION

Kenneth R. Boff, PhD
Joseph P. Martino, PhD

## 1.1 BACKGROUND

> "Endowing a computer with genuine intelligence would rank in importance with the Industrial Revolution..."
>
> FORTUNE, May 17, 1982

> "A new wave of supersmart computers is about to invade the office and the factory."
>
> HIGH TECHNOLOGY, Nov., 1982

> "Programs called expert systems are being ballyhooed as the hottest technology around. While useful for some tasks, the systems aren't as smart as they sound."
>
> FORTUNE, Aug. 20, 1984

Over the past several years, artificial intelligence or AI has surfaced in dramatic fashion in the headlines of the popular press. In turn, this has stirred the public's imagination and has fomented a good deal of speculation on the potential impact of this technology on society, at home, at the workplace and on approaches to national security. Of particular interest is the ease with which we have come to accept, or perhaps even to expect, machine intelligence as an inevitable extension of information processing technology. Given these factors, it is also not surprising that R&D managers in government and industry are rushing to gain leverage over this emerging technology by developing applications with which to drive the basic science. Hence, one purpose of this report is to provide a rational basis for assessing speculative expectations for machine intelligence by logical analysis of current and projected capabilities in artificial intelligence technology.

At present, the Defense Advanced Research Projects Agency (DARPA) is the leading government sponsor of Artifical Intelligence R&D in the United States. Their approach is centered on economically stimulating research in several conceptual application areas (e.g., Strategic Computing Program, Pilot's Associate, Autonomous Vehicles, etc.) with the goal of generating a windfall of spinoff technologies useful to national security. On a smaller scale, the Tri-services and NASA have many individual AI related R&D efforts underway, most of which appear geared towards demonstrating currently available AI tools or systems in specialized application domains. The Air Force Aerospace Medical Research Laboratory (AFAMRL) has the mandate to understand and improve the human-machine interface particularly as it evolves in the context of automation, intelligent machines, and shared intelligence systems. Hence, the main purpose of this report is to provide AFAMRL with a Technology Investment Strategy for R&D in artificial intelligence that may be rationalized within the resource structure and mission domain of the laboratory.

The preparation of this evaluation has been a costly and arduous exercise. While we have attempted to provide an objective and balanced perspective on the current state-of-the-art, it has also become evident that this is not, in all aspects, a feasible goal. The problem is that the economic incentive provided by government funding agencies and independent venture capitalists is shifting investigators away from open publication of basic science towards private capitalization on research findings maintained as proprietary. Thus, in an ironic sense, the very factors supporting the surge of investment in artificial intelligence R&D, and therefore the need for the present evaluation, has exacerbated the task of objectively assessing the current scientific baseline. As a result our attempts to achieve a comprehensive survey of the state-of-the-art of artificial intelligence technology were constrained by inaccessibility of data, our inability to verify unsubstantiated corporate claims, and the difficulty of assessing the relevance of the highly specific, sometimes obscure, basic science reported in the literature. In order to estimate risk and potential return on future technology investment, it is clear that future R&D managers will need a more reliable basis than is presently attainable for discriminating between the hype and the science of AI.

## 1.2 AIM-TECH PROGRAM PLAN

The automated information management technology program (AIM-Tech) was organized around three overlapping and interactive phases. The first phase involved selection of three technical domains within the mission domain of the laboratory to serve as foci for evaluating investment opportunities in AI. A workshop was organized to analyze the human information and control management choke points within each of these areas. Workshop participants were selected for their professional involvement in the technical domains rather than on the basis of familiarity with artificial intelligence theories or applications. The product of this workshop was used to develop a set of functional specifications for alleviating human-system interface choke points. These provide a basis for analysis in the third phase of this effort.

In phase two, Bolt Beranek and Newman Inc. was contracted to conduct an evaluation of the state-of-the-art of eight technical areas subsumed under artificial intelligence.

In phase three, an analysis was conducted of the demands on current and projected AI technology needed to alleviate the human information and control management choke points identified in phase one. A final analysis was then conducted and summarized as a Technology Investment Strategy. Each of these phases is described in greater detail below. The major sections of this report document the products of each phase.

## 1.3 PROBLEM DOMAINS

Three AFAMRL technical domains which were most amenable to application of artificial intelligence concepts and technology were selected as the foci of the Technology Investment Strategy. A general description of each is provided in the sections below.

### 1.3.1 SYSTEMS DESIGN

Data regarding the variables which impact the operator's ability to acquire and process task-critical information are of prime importance to the design of effective controls and displays. Though a considerable volume of relevant data exist in the perception and human performance research literature, these data are not in a form that can be readily accessed or interpreted by design engineers with respect to specific design problems. Access to these data is confounded by the fact that perceptual concepts on which those data are based typically lie outside the scope of the designer's previous training or experience. Identification of these concepts requires their linkage to information or issues that are familiar to the designer. While a major effort is needed to format these data so they can be understood and used by designers, access techniques based on the current state-of-the-art are insufficiently refined to enable reliable cross disciplinary access to information.

Existing information management technology is geared towards getting information in and out of an "electronic file cabinet" and not toward interfacing with the specific needs and expertise of the user. Within this existing technology, the information retrieved is only as good as the key words selected. If a keyword is too general, then the user must sort among potential "hits," "misses," and "false alarms." On the other hand, if a keyword is too specific then there is the potential that valuable information will be missed. Furthermore, the ability to effectively sort information is, in turn, dependent on the experience and training of the user. In sum, cross-disciplinary access to information may not be effectively achieved. In the long-term, the development of a next-generation computerized knowledge-based management system is needed that will aid the designer to reliably acquire and implement data relevant to specific problems.

### 1.3.2 Pilot/Aircrew Automation

The increase in the number of airborne systems and the concomitant increase in mission responsibilities for the aircrew have resulted in an increase in crew workload. For example, during hostile engagements the fighter pilot must decide what weapons to deliver, plan flight routes into and out of target areas, monitor aircraft performance, listen for radio communications, and most importantly, monitor enemy activities to prevent being shot down. One solution to this problem is to distribute workload appropriately between the pilot and an AI system such that the unique capabilities of each are being capitalized upon. Such an AI system can assimilate and accommodate new information not already within the data base and can thus assist the aircrew in a variety of mission tasks such as calculating optimal flight path in relation to threat situation, system status, mission priorities, etc., determining appropriate weapon parameters and maneuvers to be employed in response to the current threat situation, and analyzing unanticipated combinations of failures and providing solutions to the aircrew.

### 1.3.3 Command, Control, & Communication ($C^3$)

There is a current problem in both tactical and strategic intelligence centers of near real-time comparison of incoming data with prior historical data bases to assess enemy intentions. One possible solution would be to use an AI based expert system to process incoming information and recommend appropriate actions. Another important strategic $C^3$ issue is the management of weapons system status, failure diagnosis and repair aids. Military operators need an intelligent interface with complex weapons to assess operational state and alternative courses of action. Several types of expert aids are needed for mobile strategic $C^3$ facilities during protracted hostile engagement. This situation requires minimal manning and perhaps crews with varied experience, and may be an excellent application of AI technology.

## 1.4 AIM-TECH WORKSHOP

A three-day workshop was conducted during April 29 - May 2, 1984 in the three AFAMRL technical domains: systems design, pilot/aircrew automation; and command, control, and communications ($C^3$). Each technical domain was treated independently by different working groups, each of which was comprised of two group moderators, 7-10 participants, and 2-5 observer/analysts. The composition of each group was balanced in terms of the experience/perspective of the participants. (See Table 1 for a listing of participants.) The workshop used a variety of specialized techniques to stimulate the flow and exchange of ideas and information among participants with the objective of identifying control and information management choke points in each domain. Each working group was provided with a "seed problem" designed to introduce the problem and initiate the first round of discussions.

### 1.4.1 Seed Problems

- Systems Design.

  > "THE INFORMATION PROVIDED TO THE HUMAN OPERATOR AND THE WAY IT IS PRESENTED DIRECTLY REFLECTS THE DESIGNER'S UNDERSTANDING OR LACK OF UNDERSTANDING OF THE PROCESSES THAT GO ON WITHIN THE OPERATOR AS HE EXCERCISES CONTROL"

  Training simulators and operational controls and displays are typically designed without systematic consideration of data regarding human performance characteristics and limitations. The reason for this is that much of these data typically lie outside the scope of the previous training and experience of the designer. Hence, access to this potentially valuable information is constrained by a) knowledge of its existence, and b) ability of designers to penetrate the existing literature and identify data germane to their problem. Furthermore, data that are accessed may be difficult to interpret due to jargon as well as differences between design requirements and experimental conditions under which the data were collected. In sum, approaches for improving accessibility and usability of human performance data by designers are needed.

- Pilot/Aircrew Automation. Today's operational aircrews continue to experience workload saturation despite advances in display and

TABLE 1. AIM-TECH WORKSHOP PARTICIPANTS

A. SYSTEMS DESIGN

MODERATORS:

Dr. Janet Lincoln
Design Consultant
New York University

Ms. Judy Williams
Senior Human Factors Analyst
Human Factors Group
MacAulay-Brown, Inc.

PARTICIPANTS:

Mr. Ed Crowell
Senior Human Factors Analyst
E-3 Program Group
Boeing Aerospace Co.

Mr. Ed Martin
Technical Specialist, Flight and Maintenance Simulation Branch
Deputy of Engineering Trng Sys Div
AF Aeronautical Systems Division

Dr. Don Devoe
Design Consultant

Dr. Ethel Matin
Assoc Professor of Psychology
C.W. Post College
Long Island University

Mr. Tom Kelly
Deputy Branch Chief, Flight and Maintenance Simulation Branch
Deputy of Engineering Trng Sys Div
AF Aeronautical Systems Division

Mr. John Sinacori
Consulting Engineer
Pres, John B. Sinacori Associates

Dr. Robert Kinkade
Vice President, Essex Corporation

OBSERVERS/ANALYSTS:

Mr. Dave Brungart
Div Chief, Technical Services Division
AF Aerospace Medical Research Lab

Dr. Herschel Self
Engineering Research Psychologist
Human Engineering Division
AF Aerospace Medical Research Lab

Dr. Gary A. Klein
President, Industrial Psychologist
Klein Associates

Ms. Sharon Ward
Operations Research Analyst
Human Engineering Division
AF Aerospace Medical Research Lab

Mr. Mike McNeese
Engineering Psychologist
Crew Systems Division
AF Aeronautical Systems Division

TABLE 1 (Cont'd)

B. PILOT/AIRCREW AUTOMATION

MODERATORS:

Mr. Tim Anderson
Electronics Engineer
Biodynamics and Bioengineering Div
AF Aerospace Medical Research Lab

Maj Robert Eggleston
Engineering Research Psychologist
Human Engineering Division
AF Aerospace Medical Research Lab

PARTICIPANTS:

Mr. E.C. Adam
Staff Manager-MDC Fellow
Advanced Crew Systems
McDonnell-Douglas Aircraft Co.

Dr. Sheldon Baron
Division Vice President
Computer and Information Sciences
Bolt Beranek and Newman Inc.

Maj Joe Farcht
Tactical Air Command Systems Office
AF Aeronautical Systems Division

Dr. Tom Furness
Chief, Visual Display Systems Branch
Human Engineering Division
AF Aerospace Medical Research Lab

Mr. Sam Herron
Engineering Specialist
Human Factors Engineering Group
General Dynamics

Dr. William B. Rouse
Prof of Industrial and System Engineering
Georgia Institute of Technology
and
President and Principal Scientist
Search Technology Inc.

Cmd J.A. Sears
Program Manager
Info Processing Tech Office
Defense Advance Research Project Agency

OBSERVERS/ANALYSTS:

Lt Tony Aretz
Deputy Technical Director
Cockpit Automation Technology Advance Development Project Office
AF Aerospace Medical Research Lab

Mr. Reed Morgan
Deputy Project Manager
Cockpit Automation Technology Advance Development Project Office
AF Aerospace Medical Research Lab

TABLE 1 (Cont'd)

C. COMMAND, CONTROL AND COMMUNICATIONS

MODERATORS:

Capt David Leupp
Research Project Engineer
Human Engineering Division
AF Aerospace Medical Research Lab

Mr. Joseph G. Wohl
Vice President for R&D
Alphatech, Inc.

PARTICIPANTS:

Dr. Erhard O. Eimer
Professor of Psychology
Wittenberg University

Dr. Lawrence Reed
Research Psychologist
Logistics and Human Factors Div
Air Force Human Resources Lab

Dr. Mathew Kabrisky
Professor of Electrical Engineering
AF Institute of Technology

Mr. Thomas H. Thompson
President
Management Technology Associates

Lt Col Thomas Katonak
Chief, the Cheyenne Mountain Complex $C^3$ Branch
Deputy Chief of Staff, Plans
HQ Space Command

Mr. Maris Vikmanis
Electronics Engineer
Human Engineering Division
AF Aerospace Medical Research Lab

OBSERVERS/ANALYSTS:

Dr. Clifford E. Brown
Assoc Prof and Chairman of Psych
Wittenberg University

Lt Col Antone Kusmanoff
Direct of Academic Operations Support
AF Institute of Technology

Mr. Bob Centers
Project Engineer
Human Engineering Division
AF Aerospace Medical Research Lab

Mr. Don Monk
Aeronautical Engineer
Human Engineering Division
AF Aerospace Medical Research Lab

Dr. John Forester
Engineering Research Psychologist
Human Engineering Division
AF Aerospace Medical Research Lab

Mr. Walt Summers
Chief, Technology Development Human Branch
Human Engineering Division
AF Aerospace Medical Research Lab

TABLE 1 (Cont'd)

D. STAFF

| | |
|---|---|
| Dr. Kenneth R. Boff<br>AIM-Tech Program Director<br>Human Engineering Division<br>AF Aerospace Medical Research Lab | Dr. Joseph P. Martino<br>Senior Research Scientist<br>University of Dayton Research Inst |
| Mr. Gian Cacioppo<br>Manager-Human Factors Group<br>MacAulay-Brown, Inc. | Dr. Walter Reitman<br>Manager, Artificial Intelligence Department<br>Bolt Beranek and Newman Inc. |
| Ms. Gloria Calhoun<br>Engineering Research Psychologist<br>Human Engineering Division<br>AF Aerospace Medical Research Lab | Dr. Tom Rueth<br>Consultant<br>Resource Associates |
| Mr. Mark Jones<br>Human Factors Analyst<br>MacAulay-Brown, Inc. | Mr. Scott Sheely<br>President, Resource Associates |

data handling technologies. Much of the problem lies with design decisions which place a significant burden on the pilot/operator to integrate a collection of discrete, highly codified bits of information and control options into a sense of 3-space situation awareness. A factor contributing to the growth of this problem has been attempted solutions which rely solely on implementing new technologies with insufficient regard for the pilot's role and capabilities. As long as pilots remain in the control loop, in or out of the cockpit, systems effectiveness is inextricably bound to human information processing capabilities. A rational basis for design decisions regarding allocation of information management and control responsibilities is needed.

- Command, Control, and Communication ($C^3$). Processing demands on $C^3$ systems have grown exponentially since World War II because of increases in the number, mobility, and complexity of both friendly and enemy forces. To respond to the onslaught of information and shrinking time lines, system developers have tried a number of approaches. Cryptic symbology in denser formats on large and multiple displays has been used to deliver more information to each individual. In many cases, numbers of people have been increased and their numbers geographically distributed. Finally, automation has been attempted in cases where information handling and reduction was thought to be algorithmic. Using these approaches, however, raises difficult questions about operator overload, organizational complexity-vs-effectiveness, and the current quality and appropriateness of automation and about their relationship to the primary $C^3$ goal of time-constrained reduction of uncertainty and activation of an appropriate response.

### 1.4.2 Workshop Products

The documented output of each working group was distilled into a listing of functional specifications which were candidates for application of artificial intelligence based technology. These were grouped in each technical domain into three categories.

1. Communication. Approach to input/output between the operator and machine subsystems.
2. Expert understanding. Embodiment of and adaptive accessibility to domain knowledge.
3. Decision Aids. Use of domain knowledge to support control/information management operations of human operators.

Because of the apparent similarity of many of these specifications across the three technical domains, contextual scenarios were developed for each technical domain to enable differentiation among these specifications. In addition, these scenarios are intended to suggest technology endpoints, thereby providing a rough basis for projecting from present AI capabilities. These scenarios were provided to BBN for analyses of the AI capabilities necessary to support the functional technology demands of the scenarios.

## 1.5 STATE-OF-THE-ART REVIEW

The objective of this phase of the program was to develop a concise evaluation of current and emerging AI technology in the following areas:

- expert systems and knowledge engineering
- natural language
- knowledge representation
- computer vision (image understanding)
- tutoring and training
- planning and problem solving under real world conditions
- AI tools and environments
- speech

Emphasis was placed on treating expert systems and natural language because of their apparent value to the three AFAMRL technical areas. Similarly, other areas such as robotics were not directly considered because of their low apparent relevance to the technical domains. This evaluation, included as Section 2.0 of this report, treats the following factors in detail:

- state of the technology, that is, conceptual, demonstrative and fielded systems are differentiated

- probability estimates for technology transition including risks, associated costs, etc.
- major technology gaps/voids to be overcome and predictive estimates of associated time, costs, risks, etc. to accomplish this
- problems likely to remain unresolved
- major laboratories and key points of contact doing this work
- recommended key references

## 1.6 DEVELOPING A TECHNOLOGY INVESTMENT STRATEGY

This third phase of the effort involved analyses of the scenarios from the three AFAMRL technical areas to determine the AI technology requirements on which they were based. Once identified, these technology requirements were then assessed against the current state-of-the-art (detailed in phase two) and projections of emerging AI technology to determine:

- where off-the-shelf capability exists or will exist (when and with what probability) and where the Air Force can capitalize on the existing investments
- estimates of current directions of technology
- who and where AI technologies will come from
- probability estimates of success
- technical areas where success is unlikely
- where further research should be pursued

In turn, this information (detailed in Section 4.0 of this report) provides the basis for the Technology Investment Strategy (Section 5.0).

### 1.6.1 On Technology Projection

In order to assess emerging technologies for investment opportunities, it is essential to have a metric to assess the level of performance of the technology. Without such a metric, any evaluation can at best state that in some vague way the technology is better than it used to be, and that it is likely to get better yet with time. Developing a metric useful to forecasting technology entails identification of appropriate aspects or dimensions of the technology along which milestones can be charted and specific advances estimated (for detailed background on the art and science of Technology Forecasting for Decision Making, see Martino, 1983).[1]

Technology milestones were identified for each of the eight subareas of artificial intelligence and are reported in Section 4.0 of this report. These milestones were selected on the basis of the following criteria:

- Milestones must be performance-oriented and technique-independent.
  - Milestone statements represent the demonstration of a previously unachievable level of performance breakthroughs.
  - Milestone statements avoid the form "result A will be achieved using technique B," since this presents problems for both those who believe result A will be achieved by some other technique, and those who believe that technique B will not be able to achieve result A. However, a statement of the form "technique B will be brought to the stage of application, making result A possible," may be acceptable, since it leaves open the possibility of other techniques actually being used to achieve result A. The critical issue is whether it is the technique or the possibility of the result which is being forecast. Whichever it is, the forecast must focus on that instead of being "double-barreled."

---

1 Martino, J.P. (1983). Technological forecasting for decision making (2nd ed.). New York: North-Holland.

- o Each milestone must represent a discrete level of capability. The statement of a milestone should be sufficiently clear and definite that in retrospect it would be possible to determine if it had been achieved.
- o Milestones may be either quantitative or qualitative.
  - When a milestone is stated quantitatively, it should not be a point picked arbitrarily from a continuum, instead, it should represent a qualitative increase over the capability which was available before.
  - When milestones are stated qualitatively, the order of increasing performance should be apparent. That is, it should be readily apparent that each milestone represents greater performance than the one before it.
  - Qualitative milestones may often usefully be stated as thresholds passed, barriers removed, or difficulties overcome.
- o The milestone should be important from a human factors standpoint. In particular, it should represent a capability which solves some human factors problem or permits some human factors solution to be implemented.
- o The milestone should be significant in that it represents a major step forward in the usefulness of the technology.
- o The milestone should be stated in terms which make its significance for human factors evident.

### 1.6.2 AIM-Tech Investment Strategy

Our approach to defining a technology investment strategy has been to first identify those AI areas which might contribute to the solutions of Air Force problems if they were better understood. These are documented in

Section 2.0 of this report. These were matched to three AFAMRL technical domains (detailed in Section 3.0), with technology milestones identified in each AI area, and estimates of when the milestones might be achieved (detailed in Section 4.0). Hence, the investment strategy consists of a system for comparing relevant investment opportunities against the level of effort needed to achieve a technology breakthrough in each. This is detailed in Section 5.0 of this report. The main objective of this section of the report is to help assure that future funding by AFAMRL of research and development of AI technology will provide return on investment in terms of benefits of this technology to aiding the solution of Air Force problems. Data are provided to support selection of target areas for funding by providing a basis for quantifying investment decisions, thereby allowing potential value to be compared with cost.

## Section 2. Artificial Intelligence: A State-of-the-Art Review

Walter Reitman, Ph.D.
Ralph Weischedel, Ph.D.

### 2.1 OVERVIEW

This section provides a state-of-the-art review of eight areas of artificial intelligence: expert systems, natural language processing, knowledge representation, computer vision, training, planning, tools for AI, and speech. For each of those areas, the following information is provided:

- an overview
- a glossary of terms
- the review itself, including
  - operational applications
  - techniques that make effective systems
  - principal areas of current research
  - significant problems needing research
  - a list of major laboratories
  - a brief list of key references
- an executive summary

## 2.2 EXPERT SYSTEMS AND KNOWLEDGE ENGINEERING

### 2.2.1 Overview

Work in expert systems is currently the most visible area of artificial intelligence. In the last five years, initial commercial applications and the potential of revolutionary ways of using computers have spawned numerous start-up companies and even more research groups in industrial labs. In the light of such a frenzied growth period due to the widespread realization that artificial intelligence will have big payoffs, it is not surprising that the AI technology at the center of this frenzy is labeled ambiguously with the term "expert systems."

First, let's consider an inappropriately broad definition. For some, "expert systems" refers to any system that incorporates some competent decision-making, regardless of the form in which the knowledge enabling the decision-making is implemented. Thus, for example, a program that incorporates statistical capabilities might be referred to by some as an expert system. This is too broad to be useful. It is tempting for people to use it of course, since if one doesn't have an "expert" system, what one has seems highly undesirable. Is it an "inexpert" system? A "novice" system?

A second definition seems too narrow, and focuses on the manner in which the knowledge of an expert is incorporated. Many current expert systems consist of a collection of if-then rules, together with an "inference engine," namely, a procedure for applying the rules to data and previous conclusions to derive new conclusions using the rules. Additionally, the system may include some "explanation capability," which is designed to respond to questions about why it behaved in a particular way. Classically, expert systems carry out inference on the set of rules by using the rules repeatedly from the data (forward chaining), using the rules from a hypothesis to see if the data support it (backward chaining), or some combination of the two. Rules, by this definition have a form such as <u>if</u> A & B & C <u>then</u> D, where A, B, C, etc.

are conditions or facts which, if true, allow conclusion D. As with most programs, the executable version may actually be the result of a transformation, called "compilation," i.e., converting the set of rules and the inference engine into a compiled form, rather analogous to a FORTRAN compiler converting a FORTRAN program to a lower level language.

Our conclusion is that a mid-ground between the two extreme definitions is the only one that makes sense in the long term. An <u>expert system</u> explicitly incorporates knowledge based in significant part on symbolic representation of a body of facts, rules of thumb, strategies, concepts, and common sense that an expert might use in solving one of a class of problems. Such knowledge is supplemented by an inference mechanism that enables drawing conclusions from the knowledge. This is narrower than the broad definition, since it requires explicit symbolic (as opposed to numeric or equational) representation of knowledge for a significant part of the system, and since it requires that a human expert, if one exists, might reason that way. It is broader than the narrow definition by incorporating more general knowledge, such as planning knowledge, and by allowing richer representations of that knowledge as well as richer inference mechanisms when they become available. Furthermore, it allows transformation of the knowledge and inference mechanism into lower-level programming languages, as in compilation.

#### 2.2.1.1 The Knowledge Engineering Process

Since building an expert system requires substantial programming, and since the experts in general do not know AI programming, a major effort in building expert systems currently is interaction between an AI person and an expert to transform their knowledge and reasoning into programs. The process of transforming the desired knowledge and reasoning into programs (e.g., rules, an inference mechanism, explanation capability, etc.), is called "knowledge engineering." To build an expert system, one must know (a) what knowledge to incorporate, (b) what software tools are adequate for the task, and (c) how to encode knowledge and reasoning using these software tools. Typically, the knowledge engineering required for a project is very extended and complex. The proficiency of an expert system depends more on the knowledge engineering process than any other factor, for it is the encoding of the knowledge and reasoning as programs that accounts for an expert system's ability to draw conclusions.

### Understanding the Expert

The knowledge engineer must begin by building up an understanding of what the expert is talking about. There are cases, of course, in which the knowledge engineer and the expert are the same person. In general, however, the knowledge engineer must acquire sufficient understanding of the expert's domain to interpret the information provided by the expert. This is critical for maximizing the likelihood of good choices for the expert system's architecture and representation.

### Understanding the State-of-the-Art

A second requirement for the knowledge engineering process is a good understanding of the current state of knowledge-engineering techniques. Development in this field is taking place at an extremely rapid rate. On the one hand, there are new knowledge representation systems being developed. On the other, there are knowledge engineering aids, notably, the various tool kits such as KEE or ART, which may (or may not) offer substantial advantages for a particular expert system application. The knowledge engineer must understand these options if he/she is to structure his expert system effectively.

### Establishing Feasible Goals

Once the knowledge engineer has built an adequate understanding of the expert and the domain, and has considered the expert system architectures that might be appropriate to the domain, the next step is to set goals for a feasible expert system. It should not be assumed that the expert system will be able to do everything that the expert informant does. Part of the expert's skill may depend on knowledge that is difficult to express in satisfactory form in an expert system. Then, too, there will be time and resource constraints. All of these considerations imply careful planning in order to define an expert system that can be completed within the available time, using the available technology, and which, when completed, will make a significant contribution to the problem at hand.

### Selecting an Architecture and Representation

Now, with the specifications of the intended expert systems in hand, it is time to work out a detailed architecture and representation. This may employ a single knowledge representation, or it may involve a hybrid system, one that makes use of several different kinds of knowledge representations in an integrated form. (See section 2.4 for details regarding Knowledge Representation.)

### Eliciting the Knowledge Base

During the preceding stages, the knowledge engineer will have been eliciting information from the expert for preparation and planning. At this point, he must think about how to elicit a substantially complete body of information for the expert system. A number of factors are involved here, including the time available, the subject area, and considerations having to do with the personal style and preferences of the expert. In part, as a consequence of the high variability in those factors, there are no hard-and-fast procedures for knowledge engineers. At best there are some guidelines, for instance, on interviewing and on determining user requirements (Boose, 1984; Buchanan et al., 1983). In our opinion, the psychology of dealing with an individual expert in itself dashes the hope of routinizing this art. More than any procedure, the commitment of the expert and of the knowledge engineer is critical, for it involves not only the interviewing to create an initial version but also going over numerous case studies of a given version's behavior to revise the knowledge to more accurately reflect the expert's conclusions and the expert's performance. This implies much understanding by the expert of the program's basis for behavior, though not of the programming details. Furthermore, it is such thorough testing which offers the feasibility of determining the reliability of the knowledge engineering process. That knowledge engineering is an art and that it is critical to success is not a cause for alarm; rather it serves as a note of caution against blind optimism that expert systems or AI is a universal panacea.

An example of the variable nature of problems in knowledge engineering is the degree to which an expert can introspect about his/her knowledge and reasoning. The knowledge in some domains is available in reasonably discrete, rule-like form. Examples might include the kinds of knowledge that experts in insurance underwriting or in personnel functions would utilize. Compare this with the expertise involved in reading x-rays, where much of the information is visual, informal, or intuitive and, consequently, where substantial ingenuity may be required to make the rules explicit, if that is possible at all.

If the goal is to build an expert system for unsolved problems, then we have a very difficult situation; some of the problems in the Pilot's Associate, Battle Management System, and Autonomous Vehicle Systems proposed by DARPA under the Strategic Computing initiative may be of this kind. Here, since the technology is not yet available, there are by definition no domain experts. The most the expert system designer has to work with are informed guesses by experts in the closest current approximations to the technology.

## Evaluating a Prototype

Once the process of eliciting information is well along, it should be possible to begin prototyping and evaluating the initial expert system. Here is where the environments and tools associated with the LISP language are most helpful (see section 2.8). Because these environments and tools have evolved in the context of artificial intelligence programming, they include a great many aids for analyzing and modifying systems as they are being developed. The knowledge engineering process will require repeated cycles of such modification, as new information is acquired from the expert, and as evaluation indicates gaps or inconsistencies in the knowledge base.

## Building in User-Friendliness

We have not yet discussed the relation between the expert system and the user of the system. As is well known in the management information systems area, many otherwise adequate systems fail because of lack of consideration of how the user will react to the system. Thus, it is extremely important to follow up the initial knowledge engineering process with a phase in which the resulting expert system is tested for robustness and user acceptability. (Robustness includes not only a broad range of problems but the ability to respond intelligently to user behavior not necessarily envisioned by the designer.) This need dictates the use of tools for rapid prototyping, such as Interlisp, LOOPS, KEE, or other AI programming aids, and powerful LISP machines (see section 2.8 on AI Tools). Rapid prototyping is the most effective means of ensuring user acceptance, since oftentimes features of a complex system are impossible to evaluate without the user first experiencing them.

## Modifying and Extending the System

Finally, since expert knowledge changes, and the situations the knowledge is to be applied to change, the knowledge engineering process must provide for the modification and extension of the expert system. One of the claims for early expert systems was that, since they consisted of a modular collection of rules, they could be extended and modified by simply adding, changing, or deleting rules.

We now recognize that the situation is more complex than this. The problems can be even more severe when it is necessary to add new system components and more importantly, reorganize the total system. In fact, experience has shown that when these more substantial modifications are required, it may be simpler and easier to rewrite the entire expert system.

#### 2.2.1.2 Current Status of Expert Systems

Expert systems began as a spin-off from artificial intelligence, a field that was, until a few years ago, a purely academic discipline. At this point, the major producers of serious expert systems are commercial enterprises -- small start-ups, commercial laboratories, and some large industrial firms. Thus, the rules of the game regarding unconstrained information about academic research no longer apply to expert systems. In other words, the capabilities of commercial expert systems may be exaggerated, and the significant design elements that make one expert system better than another are likely to be treated as proprietary secrets. For these reasons, it is very difficult to collect detached, objective information either about how a particular commercially significant expert system works, or how effectively it works. Hence, the assessment of the technology in general, and of individual systems in particular, necessarily depends much more heavily upon word-of-mouth information from informed sources.

Below we briefly summarize some of the best known expert systems. Before we turn to specific systems, however, it is important to have some general understanding of the overall state of the field at this time.

The descriptions of specific systems which follow are mostly taken, with the author's permission, from Nickerson (unpublished manuscript). These descriptions, as well as those in the appendix, should be read with the cautions above in mind.

As Nickerson points out, there are very few expert systems in operational use. However, application areas for which expert systems are being applied or developed include computer system configuration, locomotive maintenance, oil exploration, biological research, medical diagnosis, business information management, and instruction. Among these systems are the following.

- Xcon: A system (also known as R1) used by the Digital Equipment Corporation to configure VAX computer systems in accordance with the needs and wishes of individual customers. The need for expertise comes from the fact that instead of marketing a small number of preconfigured systems, Digital offers a variety of system components (over 1000 options) from which buyers can customize systems to their tastes. Not all components are compatible with each other, however, and configurations must be designed with the knowledge of the constraints. Xcon uses about 2500 rules, and is claimed to be the largest expert system in daily use in an industrial environment anywhere in the world.

- Delta/Cats-1 (Diesel-Electric Locomotive Troubleshooting Aids/Computer Aided Troubleshooting System): This system was developed by General Electric to help diagnose problems with railroad locomotives and to facilitate maintaining them. It reportedly contains over 500 "if...then" rules, runs on a PDP11/23, has 10 megabytes of disk memory and uses a VT100 terminal and a Selanar graphics board. It also contains a video disk player, which allows the system to provide the user with drawings, photos and movies as appropriate.

- Prospector: This was one of the earliest expert systems. It analyzed data to determine likely sites for ore, such as porphyry copper deposits and molybdenum.

- DipMeter Advisor: Developed by Schlumberger Ltd. for analysis of oil well drilling data, the DipMeter Advisor gets its name from the fact that one objective of the system is to determine the angular displacement, or "dip" from the horizontal, of subsurface mineral strata. Its purpose is to help geologists interpret data obtained from a dipmeter inserted into drill holes. This system is claimed to now be undergoing extensive field testing.

- Drilling Advisor: The Drilling Advisor was developed jointly by Teknowledge Inc. and the French National Oil Company Societe Nationale Elf Aquitaine. Its purpose is to provide consultation to the supervisor of an oil rig regarding the problem of "sticking," which is often encountered in the drilling of production oil wells. Sticking refers to a situation in which it is impossible either to continue drilling or to raise the down-hole equipment to the surface. The Drilling Advisor is intended to help diagnose the most likely causes of such problems, and to recommend actions aimed at alleviating or avoiding them. Its knowledge base contains about 250 if-then rules.

  In diagnosing a problem, the Drilling Advisor attempts to identify the most likely of six possible causes of sticking. It qualifies each hypothesized diagnosis with a probability reflecting its degree of certainty. Diagnoses are accompanied by explanations of the reasoning on which they are based. Prescribed treatments are also selected from a relatively small set of possibilities. In diagnosing, the system requests information from the user regarding the well, constituent rock types, type of activity immediately preceding the sticking, depth of drill bit, and so on. When it has proceeded far enough to form a tentative hypothesis, the specific questions it asks are contingent on that hypothesis.

Elf Aquitaine has made positive statements about the system. However, Elf has an equity position in Teknowledge. Other sources give varying reports about the system's effectiveness.

- Puff-VM: Developed by Stanford University and the Pacific Medical Center, Puff is a small production-rule system for helping to diagnose lung disorders. It takes a patient's history and a variety of measurements and test results as inputs and produces a diagnosis, which is added to the patient's records and is checked by a physician.

- Mycin: Also developed at Stanford University, Mycin was intended to assist in the diagnosis and treatment of infectious diseases and in the selection of antibiotics appropriate to their treatment. Mycin's data base contains about 500 rules in the form of if-then statements. In attempting a diagnosis, Mycin tests the various rules in its data base against information that has been provided about the patient.

  Mycin has a limited ability to explain to the user at least some aspects of its processes. If, for example, the user types "why" in response to a request from the program for additional information, the system responds with an explanation of why it wants the information requested. The explanation reveals the rule that it is currently working on and why it is working on that rule. By typing <u>why</u> repeatedly, the user can back the system up through its entire chain of inferences. This feature adds to the usefulness of the system for purposes of training.

- Internist-1: Developed at the University of Pittsburgh, Internist-1 is intended to assist in diagnosis in internal medicine. Its diagnostic capability was intended to be broader than that of previously developed systems and to apply to the diagnoses of multiple and complex disorders. The inferential methods it uses to arrive at a set of possible diagnoses and to select the most appropriate alternative from among that set were modeled after those that are believed to be used by physicians when confronted with similar diagnostic problems.

  The knowledge base of Internist-1 represents 15 person-years of work, contains over 500 disease profiles, approximately 3550 disease manifestations (symptoms), and about 6500 relations among manifestations (information regarding how the presence or absence of a given manifestation may influence the presence or absence of other manifestations). Associated with each manifestation in a disease profile are an evoking strength (the degree to which that disease explains that manifestation) and a frequency (the frequency with which patients with that disease have that manifestation); also associated with each manifestation is a disease-independent import (the extent to which the manifestation requires an explanation). Diagnoses are produced by application of a scoring procedure involving assigning numerical values to evoking strengths, frequencies, and imports and combining these values in accordance with a set of ad hoc heuristics.

Internist-1 is still viewed by its originators as a research tool, and much of their current work is focused on identifying its specific shortcomings and limitations for the purpose of paving the way to the development of more effective systems.

- Steamer: A graphics-oriented system developed at BBN for training operators of a steam propulsion plant. The system contains a model from which it can generate graphical representations of the plant, or components thereof, at different levels of detail. It can also represent graphically the flow of water or steam through the system and the consequences of specific malfunctions. It permits structured tutoring in which it presents problems to the student and guides the session, and also exploratory learning whereby the student can perform "what if" experiments and thus discover the consequences of various operator actions.

### 2.2.2 Glossary

Backward chaining: Reasoning backward from desired conclusions.

Chaining: Using rules one after the other to draw a complex conclusion in several steps. For instance, if we have two simple rules, *if* A & B *then* C and *if* C & D *then* E, we can conclude E if A, B, and D are true. The rules may be used in forward chaining by drawing conclusions from the data; namely, A, B, and D being true would give us two conclusions; C and E. Alternatively, if we hypothesize that E might be true, we might use the rules via backward chaining to determine whether the data supports it, namely, if A, B, and D are true.

Contexting mechanisms: Grouping related rules together, to reduce search.

Forward chaining: Reasoning forward from what is initially given.

If-then rule: A pairing of a situation specification with some action to be taken if that situation occurs.

Inference engine: A component that carries out the action specified by a rule, altering the situation accordingly.

Knowledge acquisition: Extracting expert knowledge from the expert and adding it to an expert system's knowledge base.

Knowledge engineering: The process of translating the knowledge and reasoning of an expert into computer programs. Since normally the domain expertise and AI programming do not reside in the same individual, this normally involves intense cooperation by at least one expert and at least one AI programmer to build an expert system.

Knowledge refinement: The process of adding and modifying rules in the rule base.

Production rule: Another name for an if-then rule.

Rule packets: Collections of related rules grouped together by a contexting method.

### 2.2.3 State-of-the-Art

#### 2.2.3.1 Operational Applications

As of September, 1984, the number of fully operational expert system applications in regular use under field conditions is probably no more than ten.

The best examples of heavily used operational systems are the two Digital Equipment expert systems, XCON (formerly called R-1) and XSEL. In addition, the Puff Pulmonary Analyzer is allegedly in use on a regular basis for analyzing pulmonary disorders.

Several other systems are in the advanced field test stage. These include AT&T's Ace system, which diagnoses, locates, and schedules repair of phone cable malfunctions; and the dipmeter advisor system being developed by Schlumberger-Doll.

There are 100-200 other expert systems that have been described as in some stage of development. A good description, overview, and characterization of the state of this collection of expert systems as of mid-1982 is Gevarter.

#### 2.2.3.2 Techniques that Make for Effective Operational Systems

Since there are so few operational systems, and since those that exist are mostly proprietary, it is difficult to do more than make informed guesses. However, we would conjecture that the following properties increase the likelihood of an effective operational system.

- The subject matter may already be structured naturally as highly codified rules. Examples might be the rules governing interest payments and charges on bank accounts and certificates of deposit. Of course, this greatly simplifies the knowledge engineering process, since the subject matter is naturally near a usable representation.

- The description of the situation given as input for the expert system may be representable as a collection of properti^ Many medical diagnosis problems have this property; for instance, the symptoms and test results form a collection of properties regarding the patient.

- An expert system may be decomposable, i.e., the set of rules may be broken into contexts or subsets of rules, with each subset appropriate to a particular state of the process. The expert system XCON is decomposable.

- There may be many acceptable solutions to any given input problem. This of course, may simplify the search, since any acceptable solution may be adequate. This is another property true of the domain of the expert system XCON.

- No reasoning may be required based on a complex model of some operating mechanism nor based on experience that happens to be difficult to analyze. Interpreting dipmeter data has this simplifying property. Of course, some mechanisms and some experience are easy to model.

We say an expert system is decomposable when it is possible to break the set of rules into contexts or rule packets, with each set appropriate to a particular stage of the process.

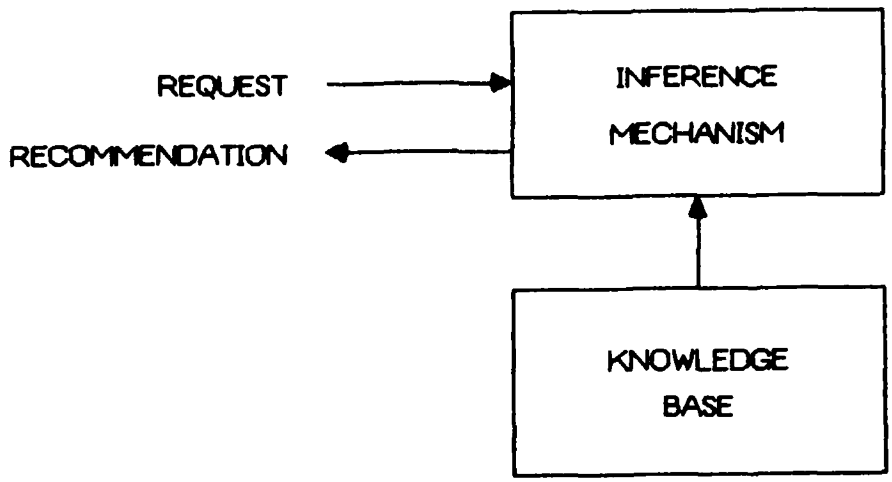
Note, finally, that the expert system has to be integrated as a component into the overall processing system in order for it to be effective. Figure 1 shows the typical architecture of current expert systems; they may be embedded in a larger application system, could in the future have a natural language interface, or could interface directly with the user in an artificial language as a present.

#### 2.2.3.3 Principal Areas of Research

The following principal areas are currently under investigation in expert systems:

- Automatic procedures for inducing rules from data. This would particularly be helpful in reducing the effort when experts have trouble introspecting about decision making or in situations where no expert is available.

- Increasing the expressive power of the rule formalism (primarily with respect to time-oriented data, and causal information). Knowledge representation techniques are weak in those areas, thereby limiting the problems to which expert systems may apply.




**Typical Expert System Architecture**

(All arrows indicate data flow.)

**Figure 1.**

- o Developing effective tools for diagnosing errors or incompleteness in the rule set, and assisting the user to modify/correct these appropriately. Such "debugging" is exacerbated with large rule sets, the unfamiliar control structure of inference mechanisms, and the degree of detail the expert must specify.

- o Better methods for dealing with uncertain, incomplete, and erroneous input information. Many applications imply such input by the nature of the problem; techniques for reasoning in such conditions are a fundamental need.

All of this work is in the research stage. No techniques for dealing with these problems have emerged as yet; rather progress is being made principally by case studies of building individual expert systems. However, each of these research problems appears to be manageable, and we anticipate limited success within five years. It should be noted that if this research is successful and these more powerful systems are developed, there is a tradeoff between the complexity of these systems and their cost in terms of computational resources required to run the systems and human effort in creating them. All known problems are being pursued at some level, though some of the research may be classified as knowledge representation, planning, or natural language processing when appearing in conference proceedings, etc.

#### 2.2.3.4 Major Gaps and Problems

The gaps targeted by the DARPA strategic computing initiative are shown below. Each of these is being addressed as a result of that initiative. Section 4 of this report itemizes research problems not currently supported.

- o More flexible control structures are needed than simply backward chaining or forward chaining.

- o More powerful representation techniques are needed, for instance, to adequately encode knowledge about time, space, and causality. Section 2.4 on Knowledge Representation amplifies this issue.

- o Aids to knowledge acquisition are needed, since acquiring the knowledge of an expert and encoding it in programs is the most difficult problem in knowledge engineering.

- o The input may contain uncertainties, errors, incompleteness, or misinformation. Obviously, this is a key in adversarial situations. Ignoring disconfirming data, for instance, is not reasonable, since that data may be the key to rejecting a wrong hypothesis.

- "Fusion" refers to the ability of an expert system to combine information from a variety of sources.

- "Explanation" is a term that has been used to describe the ability of expert systems to respond to "why" and "how" questions. This is a very much weaker and more limited form of explanation than those that can be provided by a human expert. It is generally agreed that the limited explanatory capabilities of current expert systems, though useful, need to be expanded if these systems are to be entrusted with substantially greater responsibilities and more complex tasks. A further word on the problem of "explanation" may be helpful. The problem has to do with the differences between what the system can tell the user and what the user wants to know. This is particularly clear in the case of much of the work on medical diagnosis systems. These systems are presently not utilized on a regular basis. Partly this is because, although they allegedly contain most of the information that is relevant to making a diagnosis, the explanatory mechanism is inadequate. These systems do not allow the users to query in unconstrained ways. Consequently, the medical team members do not develop the confidence in the systems necessary to be willing to use them to make important decisions. This relates back to the need to integrate expert systems into their larger decision-making context. This also restricts our ability to subject such systems to extensive tests: because they are not fully integrated into a decision-making context, they cannot be put to a complete test.

- The need for multiprocessor architectures derives from the requirement for a higher rate of processing expert system rules. In addition, multiprocessor architectures with appropriate operating systems might enable us to explore several potential solution paths at the same time, thereby greatly increasing the real-time operating effectiveness of systems.

- Expert systems originally were intended to enable computers to carry on some of the nonnumeric information processing characteristics of human experts. Now, efforts are being made to apply the same technology to the design of systems which will be capable of sophisticated decision making in the absence of existing experts. This is particularly true in some of the military applications that DARPA is funding under the strategic computing initiative. It should be clear that all of the payoffs of such systems, if they can be designed, will be high, but that building such systems entails substantially greater problems than building systems that can make use of existing experts as models.

Additionally, we should note that all current systems are targeted at specific problems. We do not yet know how to build systems that can evolve dynamically and adaptively respond to changes in problem situations. One of the putative advantages of rule based systems is their modular structure, which supposedly permits easy modification. It remains to be proven whether this ease of modification will be true in more complex systems.

#### 2.2.3.5 Approaches That Failed

There really have been no outright failures in the short history of this area. Rather, there are gaps to be filled, such as the ones listed in the previous section.

Furthermore, it is clear that just having a large LISP (or whatever language) program does not mean one has an expert system. Rather, as our definition states, what is critical is an explicit, symbolic representation of knowledge, and reasoning processes similar to what an expert might use, for a substantial portion of the system. This is critical for good software engineering (e.g., initial construction and maintenance) even if the knowledge and reasoning are compiled.

#### 2.2.3.6 Major Laboratories and Key Contact Points

- Carnegie-Mellon University, Pittsburgh, PA/John McDermott, Mark Fox
- Rutgers University, New Brunswick, NJ/C. Kulikowski, S. Weiss, T. Mitchell
- MIT, Cambridge, MA/Randall Davis, Charles Rich
- BBN, Cambridge, MA/Al Stevens, N. S. Sridharan
- IntelliCorp, Palo Alto, CA/Thomas Kehler, Richard Fikes
- Tecknowledge, Palo Alto, CA/Rick Hayes-Roth
- AT&T, Murray Hill, NJ/G. Vesonder
- Stanford University, Stanford, CA/E. Feigenbaum, W. Clancey

- Syntelligence, Menlo Park CA/Richard Duda

- *USC/Information Sciences Institute*, Marina del Rey, CA/W. Swartout

- Other Laboratories: Schlumberger-Doll, Ridgefield, CT; Fairchild, Palo Alto, CA; SRI International, Menlo Park, CA; Hewlett-Packard, Palo Alto, CA; and Xerox PARC, Palo Alto, CA

### 2.2.3.7 Recommended Key References


Aikins, J. Prototypical knowledge for expert systems. Artificial Intelligence, 1983, 20, 163-210.

Boose, J.H. Personal construct theory and the transfer of human expertise. Proceedings of AAAI-84, Austin, TX, 1984, pp. 27-33.

Buchanan, B. Research on expert systems. In J. Hayes, D. Michie & Y. Pao (Eds.), Machine Intelligence. New York: Wiley, 1982.

Buchanan, B.G., Barstow, D., Bechtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T., & Waterman, D.A. Constructing an expert system. In F. Hayes-Roth, D.A. Waterman, & D.B. Lenat (Eds.), Building Expert Systems. Reading: Addison-Wesley Publishing Company, Inc., 1983.

Buchanan, B., & Duda, R. Principles of rule-based expert systems (Department of Computer Sciences Technical Report). Stanford: Stanford University, 1982.

Campbell, A., Hollister, V., Duda, R., & Hart, P. Recognition of a hidden mineral deposit by an artificial intelligence program. Science, 1982, 217, 927-929.

Chandrasekaran, B. Expert systems: matching techniques to tasks. In W. Reitman (Ed.), Artificial Intelligence Applications for Business. Norwood: Ablex Publishing Corp., 1984.

Duda, R., & Reboh, R. AI and decisions making: the prospector experience. In W. Reitman (Ed.), Artificial Intelligence Applications for Business. Norwood: Ablex Publishing Corp., 1984.

Duda, R., & Shortliffe, E. Expert systems research. Science, 1983, 220, 260-268.

Gevarter, W. An overview of expert systems (Report No. NBSIR 82-2505). Washington, D.C.: National Bureau of Standards, October 1982.

Hayes-Roth, F., Waterman, D., & Lenat, D. (Eds.). Building Expert Systems. New York: Addison-Wesley, 1983.

Nickerson, R.S. Information Systems and People, unpublished manuscript. Cambridge, MA: Bolt Beranek and Newman, Inc., 1984.

Politakis, P., & Weiss, S. Using empirical analysis to refine expert system knowledge bases. Artificial Intelligence, 1984, 22, 23-48.

Pople, H. Knowledge-based expert systems: the buy or build decision. In W. Reitman (Ed.), Artificial Intelligence Applications for Business. Norwood: Ablex Publishing Corp., 1984.

Reitman, W. (Ed.). Artificial Intelligence Applications for Business. Norwood: Ablex Publishing Corp., 1984.

Weiss, S., & Kulikowski, C. A practical guide to designing expert systems. Totowa, New Jersey: Rowman & Allanheld, 1984.

Winston, P., & Prendergast, K. (Eds.). The AI Business: Commercial Uses of Artificial Intelligence. Cambridge, MA: MIT Press, 1984.

2.2.4 Summary

Some applications have already proven commercially viable; the recent frenzied growth in AI start-up companies and industrial research labs testifies abundantly to that. The criteria regarding whether an application problem is likely to yield to expert system technology are unclear; e.g., it is doubtful that one could design a useful expert system to give advice on the appropriateness of expert systems technology for a given problem. Expert systems thus far have been developed only in applications where one or more experts can employ introspection about their decision making. Few current systems run in real-time.

Table 2 provides a brief summary of some expert systems that have been placed in use or have received extensive research and development effort.

TABLE 2. SOME EXTENSIVELY DEVELOPED EXPERT SYSTEMS

| Name | Application | Organization | Contact |
|---|---|---|---|
| ACE | Analysis of telephone cable trouble spots | Bell Laboratories Whippany, NJ | Gregg Vesonder |
| CADUCEUS | Internal medicine | Univ. of Pittsburgh | Harry Pople; J. Myers |
| CASNET | Consultation regarding glaucoma treatment | Rutgers University | C. Kulikowski |
| DELTA (formerly CATS-1) | Troubleshooting diesel locomotives | General Electric | Francis Lynch |
| DENDRAL | Projecting molecular structure from mass spectrograms | Stanford Univ. | Edward Feigenbaum; Joshua Lederberg |
| DIPMETER ADVISOR | Interpreting oil well drilling log data | Schlumberger-Doll | Howard Austin |
| MDX | Medical diagnosis for cholestasis | Ohio State Univ. | B. Chandrasekaren |
| MYCIN | Diagnosis and treatment of bacterial infections of blood | Stanford Univ. | Edward Shortliffe |
| PROSPECTOR | Predicting likely ore deposits | SRI Int. | R. Duda (now at Syntelligence) |

| | | | |
|---|---|---|---|
| PUFF | Consultation regarding pulmonary disorders | Stanford University | J. Kunz |
| STEAMER | Training regarding operating of a steam propulsion plant | BBN Labs. | Al Stevens |
| XCON (formerly R1) | Configuring VAX computers given a customer order | Digital Equipment Corp. | J. McDermott |

Though a potentially large class of applications of current technology is amenable to commercial success, significant advances in knowledge representation, planning, and natural language processing seem necessary to broaden the class of operational applications.

## 2.3 NATURAL LANGUAGE

### 2.3.1 Overview

The goal of this work is to enable computers to communicate in natural language. By this we mean that they will understand normal communications that humans use with one another and will be able to respond to them appropriately. Since the special problems of speech input/output are covered in a separate section, we will assume here that communication between human and machine takes place through an alphanumeric terminal. "Natural language" includes not only polished prose; but also spontaneous, sometimes ill-formed utterances; jargon; and specialized forms as in chemical formulas or in some highly formatted military messages. Natural language communication involves both understanding (input) and generation (output), which so far have generally been studied separately.

There are reasons why natural language understanding is desirable:

- o some input would be available in no other form, such as newspaper articles or the comments field of even what is otherwise a very stylized, constrained, military message.
- o it obviates the need for consciously translating requests into an artificial language. This is particularly critical if the individual should be focusing on other tasks, as in the case of a pilot.
- o for an infrequent user, the idiosyncratic detail of an artificial language will be a source of frustration or will be a barrier, since remembering the morass of detail is unlikely unless frequently used.
- o even frequent users have facilities which they use infrequently and therefore for which natural language will prove convenient.
- o artificial languages tend to require great precision; nevertheless, sometimes it seems almost impossible to be that precise, as in requesting help when one is at a loss. Typical online help facilities suffer from this.
- o natural language conveys vast amounts of information concisely. For instance, if one says to a train conductor, "Culver City?," the conductor answers correctly without the need to spell everything out,

as in "Does this train stop at Culver City?"

There are also straightforward reasons for wanting natural language generation:

- o explanation, appropriate to the understanding of the user, seems critical in knowing whether to follow the advice of an expert system, to supply additional information to it, or to consult another expert
- o paraphrasing the system's understanding of a user's requests/input is critical to make sure no miscommunication is occurring and to clarify what the user wants in light of vagueness or ambiguity
- o as in the case cited for understanding, natural language output can be marvelously concise for conveying certain information, just as graphs, tables, or pictures are ideal for other data

Since programming language technology is so advanced, why isn't natural language a present capability of computers? One reason is already evident in the example above of the cryptic dialogue with the train conductor. Namely, context external to the language itself will normally have a significant effect on the interpretation of the communication. Second, ambiguity not only occurs, but is common in natural language; context determines what is intended. For example, in "Display all malfunction reports on planes in squadron 45 and in squadron 43," one wants reports on two squadrons. However, in "Display all planes that were in service in January and in February," one could want to know only about the ones in service in both months or alternatively about those in service in either month. Third, though there is much success in interpreting programming languages, there is little success to date in computer generation of meaningful expressions in either artificial or natural languages.

Effective communication entails integration of the following broad collection of capabilities:

- o understanding the content of a single sentence, on a sentence by sentence basis. If one cannot extract the meaning of a sentence in isolation, there is no basis for answering questions, carrying out requests, etc.

- o understanding the user's intentions and plans. Without this, one encounters humorous (or frustrating) situations because of purely literal interpretations, such as being answered "yes" to the question, "Can you pass the salt?"

- o understanding discourse structure. Plans are usually complex, multi-faceted structures revealed over several sentences. Modeling the structure of the discourse has proved critical to machine understanding of user intention, use of descriptions, and meaning of cryptic language.

- o dealing with ill-formed language. Typed or spoken language has a high frequency of ungrammaticalities, fragments (rather than sentences), spelling errors, slips of the tongue, etc. Such forms are termed ill-formed and provide a particular problem for machine language understanding since the rules of well-formed language have proven a key to determining what is meant.

- o knowing how to clarify or even correct misunderstandings. Misunderstandings occur even among native speakers of a language. Therefore, how much more important if we command computers via natural language that they be able to recognize and clarify the situation when potential misunderstanding arises.

- o interacting with the user in graphics and language the user can understand. The alternative *does not bear consideration.*

- o understanding how to assist the user with his/her task. Sometimes even an expert user needs help, such as what to do next, knowing how to communicate what he/she wants, etc.

Of those capabilities only the problems of sentential syntax are generally well understood. That is, research in natural language has had ten years of experience with systems that can look at the sequence of words in a sentence and determine the syntactic function of each of the sentence components. In each of these other areas, research is under way, but we are a long way from understanding how to build natural language systems that incorporate these capabilities in an effective general fashion.

### 2.3.2 Glossary

Anaphora: reference to something earlier in the communication. Pronouns (like "he"), definite noun phrases, (like "the big dog"), and demonstratives (like "this" and "that") can be used in this way.

Deixis: referring to something implied from extralinguistic context, e.g., the observable environment, rather than from the previous text. Pronouns, definite noun phrases and demonstratives can be used in this fashion. "That," when accompanied by pointing to an object on a map, in "That's the objective" is a deictic reference.

Discourse: large linguistic units consisting of connected sentences, paragraphs, dialogues, etc.

Ellipsis: a fragment which in context expresses a complete thought. For example, one can answer the question "Did you go to Chicago?" with the elliptical form "Last month," which in context means "Last month I went to Chicago."

Grammar: a body of rules describing the structure and meaning of well-formed phrases, such as words (morphology), noun phrases, and sentences. A grammar for spoken language also specifies phonological rules, describing the acoustic realization of the phrases of the language. (For written language, there are rules for spelling and punctuation instead.) The word "grammar" is sometimes used in a broader sense, when one talks about developing grammars for discourses or stories, rather than sentences.

Natural Language: any of the languages normally spoken by humans, e.g., English, Swahili, Japanese, etc.

Parsing: the process of taking a sequence of words, usually a sentence, and determining what its syntactic structure is. A parser is an algorithm for parsing a sequence of symbols to determine the corresponding syntactic structure.

Pragmatics: the branch of linguistics which describes the actual use of language, rather than the structure of language (described by syntax) or the meaning of language (described by semantics). Pragmatics deals with the conventions among speakers about how language is used to convey intention and meaning. Pragmatics also describes how the intended meanings of utterances depend upon the real world contexts in which they are uttered.

Semantics: the branch of linguistics which describes the meanings of words, sentences, and larger discourse units such as paragraphs or whole conversations. This involves the specification of rules for deriving the meaning of a sentence from the meanings of its word and phrase elements, given the syntax of the sentence. At the discourse level, semantic rules build higher order structural representations that express not only the meanings of the individual sentences, but also the meaningful relations among the sentences. This may involve interpreting pieces of discourse as speech acts in terms of the speaker's intentions, plans and goals.

Speech Acts: social acts which are performed by uttering a sentence or discourse unit. Promises and requests are forms of speech acts. A speech act has two components:

1. Its illocutionary force (e.g., asking a question, making a statement, making a promise, etc.)
2. Its propositional content (the description of what is asked, stated, promised, etc.)

Syntax: the rules of a language which describe how words can be combined to form larger linguistic entities, such as phrases, clauses and sentences. The syntactic rules also specify the internal structure of the entities which are built up in this way.

### 2.3.3 State-of-the-Art

#### 2.3.3.1 Operational Applications

There are several commercially available pseudo-natural-language systems on the market. However, none of them can be said to be "operational" in the sense that you can give it to a naive user and expect it to produce reliably meaningful and relevant answers to questions. In the hands of a user who understands the limitations in such systems, they can be said to be operational in a limited sense. The main examples of such systems are Intellect and Themis.

Intellect is produced by the Artificial Intelligence Corporation (AIC) in Waltham, MA. It sells for $70,000 and operates in an IBM mainframe environment. It has been licensed to Cullinet Software (under the name Online English), Information Sciences (as GRS Executive), and IBM. Intellect was the first system on the market. However, the natural language component of Intellect is based on decade-old technology and has serious problems in resource use (both space and time).

A major installation at Atlantic Richfield Corporation is underway that will make Intellect available to over 200 users at 10 sites. AIC is expected to introduce a version of Intellect that runs on an IBM PC XT which then interfaces with a mainframe computer that houses the data base management system. Future improvements will also include interfacing with various spreadsheet and report generator systems.

Themis is a product of Frey Associates in Amherst, NH and is currently behind schedule in beta-testing, i.e., experimental use of software outside of the site where it was created. It is priced at $24,000, interfaces to two relational data base management systems (Datatrieve and Oracle), runs on DEC VAX-11 minicomputers and requires about 2M bytes of memory. It does not have the graphical output capabilities of Intellect, but is reported to be more efficient.

Mathematica Products Group in Princeton, NJ, recently introduced a system called English which sells for $24,000 and interfaces to their Ramis II query system. However, this "English" system cannot even handle verbs. A Datamation article reported that instead of saying "Show me all the cars that went to California," the user must phrase the query to reflect the fields of the data base: "Show me the cars with shipper state Pennsylvania and destination state California."

About to enter the home and small business market is Symantec of Sunnyvale, CA, which is hoping to begin marketing a natural language interface integrated with a data base system some time in 1985. The package will run in Pascal on an IBM PC with 256K bytes of memory and a hard disk drive. (The company was originally expected to have a product on the market nearly a year ago; they have had considerable difficulty defining a product and squeezing it onto a microcomputer.)

Additional sources in this area include Texas Instruments' Natural Link (a menu-based data base management system interface that allows the user to compose a sentence by choosing from a limited set of words and phrases displayed in menus on the screen), Cognitive Systems' custom-built natural language interfaces, and Excalibur Technologies' Savvy (which can run on personal computers and uses a pattern-recognition scheme).

There are also several advanced demonstration systems available, notably, the BBN IRUS system. These utilize more sophisticated technology, and therefore provide a stronger base for incorporating results of current and future research.

### 2.3.3.2 Techniques that Make for Effective Operational Systems

Natural language understanding systems succeed best when they deal with concrete, reasonably well defined, reasonably easily symbolized areas of conversation. Much of human conversation has to do with properties of the real world, or properties of human experience, feelings, etc. These are things that people are well qualified to gain experience in, but where we don't know how to provide equivalent experience to computers. Thus, it is very difficult to provide the semantic basis for a natural language understanding system that would enable it to communicate about such areas.

It is fairly generally agreed that all sources of knowledge are critical to understand and generate natural language. These sources include:

- vocabulary
- grammar
- a knowledge representation language (this is discussed in the next chapter)
- a tightly scoped and restricted domain (e.g., a particular data base)
- a knowledge base for this domain (e.g., the facts)
- inference methods
- models of linguistic and extra-linguistic context, e.g., user goals and beliefs, entities in context, etc.

Note that the need for a knowledge representation language, a restricted domain, a knowledge base, and an inference mechanism were critical for the success of expert systems as well.

Several grammar formalisms exist, and these imply techniques for vocabularies (more formally called lexicons). Examples are lexical functional grammar, augmented transition networks (ATN), unification grammar, and augmented context-free grammars. Winograd (1983) provides an in-depth survey.

There are no unified techniques at present for modeling and using linguistic context. Joshi et al. (1981) and also Brady and Berwick (1983) contain a number of recent papers in this research area.

#### 2.3.3.3 Principal Areas of Research

One major focus is higher-order linguistic phenomena, trying for a more complete understanding of discourse. This involves understanding references to entities implicit or explicit in previous parts of the discourse (anaphora) and also references to entities in extralinguistic context (deixis). It also involves building models of user intentions, their goals, and plans.

Another major line of work is trying to extend natural language systems to the point where they can deal with ill-formed input (i.e., input involving deviations from strict grammaticality).

Additional work is going on in broadening and strengthening syntactic and semantic capabilities. There is much that is not understood, such as semantics for vague terms and significance of particular syntactic constructions.

Finally, there is substantial interest in natural language generation, i.e., getting a component to produce coherent, comprehensible discourse, as well as understand it.

All four of these are basic research areas, with some limited prototype systems illustrating possible solution procedures. There are no fundamentally insoluble or problematic issues associated with any of these areas, so we can anticipate at least limited success in the long run with probability .9 or better. The major bottleneck is the time and effort involved in modeling increasingly broad and complex subject domains.

Assuming that current trends and current work as outlined above are successful, it is likely we will be able to produce a system that can understand substantial amounts of human conversation. It would function as a very literal-minded, narrow, but nonetheless, quite useful assistant that can communicate with us.

### 2.3.3.4 Major Gaps and Problems

Besides the areas listed above, there are three additional problems. One is the design of generation and understanding components so that a system can understand what it says and vice versa. The two areas have been studied separately thus far, since each has rather unique aspects.

The more problematic areas have to do with the use of metaphor, and other more "creative" uses of language, to express new meanings or to extend or vary an accepted meaning of a term in a new way. Additionally, nothing in ongoing work will enable systems to understand more personal self-expressive meanings of language, rhetorical uses of language, etc.

Another open area has to do with the relation between purely linguistic meanings, and meanings that are tied to extralinguistic context. At the moment, our ability to design systems that are capable of ascertaining the extralinguistic context directly, without a human intermediary, is extremely limited. This involves questions of machine perception that are not dealt with in this report.

Though not a problem in natural language per se, it should be pointed out that natural language research and knowledge representation are synergistic. Timely progress in natural language certainly assumes adequate progress in knowledge representation.

The one problem not likely to be pursued in the short term, say, within the next three years, is natural language across domains, rather than over a single narrowly defined domain. See section 4.3.3 for a projection on this.

2.3.3.5 Alternative Approaches

There have been other approaches to building natural language understanding systems---syntax-free semantics (Schank & Riesbeck, 1981); semantics-free syntax; keyword analysis; and various kinds of mathematically-based models, (e.g., Markov models of natural language). With the exception of Schank's group at Yale, and some of his students, nobody believes these approaches to be adequate. Instead, the general sense of most researchers in this field, is that it takes all sources of knowledge (vocabulary, syntax, semantics, and pragmatics) at the very least as a basis for an adequate natural language system. Any attempt to leave out one of these major components results in loss of capability compared to human understanding and use of natural language utterances.

For instance, keyword analysis seems appropriate only for tasks of message routing, i.e., determining who receives a message, or for broad bibliographic search. Syntax-free semantics seems appropriate only for tasks where superficial analysis is adequate without understanding of everything. For instance, in a data base environment, the only way to distinguish between the following two requests is by syntax (which conveys the intended semantics).

- List all assets of any company that were sold to XYZ in 1984.
- List all assets of any company that was sold to XYZ in 1984.

Similarly, semantics-free syntax is inappropriate where reliable understanding is required, for syntax alone is insufficient to understand that:

- time flies like an arrow

has one meaning rather than four. Nevertheless, it could be useful in tasks of purely stylistic feedback to authors editing their manuscripts.

#### 2.3.3.6 Approaches That Failed

In some sense, no approaches have failed totally. Nevertheless, any approach that does not employ all sources of knowledge (vocabulary, syntax, semantics, and pragmatics) is bound to be severely limited. See the previous section on alternative approaches, where those limitations are illustrated.

#### 2.3.3.7 Major Laboratories and Key Contact Points

- SRI International, Menlo Park, CA/Barbara Grosz, Jerry Hobbs, Raymond Perrault, Stan Rosenschein, Jane Robinson
- BBN Labs, Cambridge, MA/Lynn Bates, Rusty Bobrow, Remko Scha, Candy Sidner, Ralph Weischedel
- University of Pennsylvania, Philadelphia, PA/Aravind Joshi, Bonnie Webber, Tim Finim
- USC/Information Sciences Institute, Marina del Rey, CA/William Mann, Norman Sondheimer
- Yale University, New Haven, CT/Roger Schank, Chris Riesbeck
- University of Texas, Austin, TX/Robert Simmons
- New York University, New York, NY/Ralph Grishman, Naomi Sager
- Philips Research Laboratories, Eindhoven, Neth./Jan Landsbergen
- MCC, Austin, TX/Jonathan Slocum
- University of Massachusetts, Amherst, MA/David McDonald, Wendy Lehnert
- Burroughs Corporation, Paoli, PA/Lynette Hirschman, Martha Palmer

#### 2.3.3.8 Recommended Key References

Allen, J. (Ed.) Special issue on ill-formed input. American Journal of Computational Linguistics, 1983, 9, 3-4.

Bates, M. & Bobrow, R.J. Natural language interfaces: what's here, what's coming, and who needs it. In W. Reitman (Ed.), Artificial Intelligence

Applications for Business. New York: Ablex Publishing Corp., 1984.

Brady, M. & Berwick, R.C. (Eds). Computational Models of Discourse. Cambridge, MA: The MIT Press, 1983.

Cercone, N. (Ed.). Computational Linguistics. Oxford: Pergamon Press, 1983.

Charniak, E. & Wilks, Y. (Eds.). Computational Semantics. New York: North-Holland Publishing Co., 1976.

Johnson, C. & Bachenko, J. Applied computational linguistics in perspective: proceedings of the workshop. American Journal of Computational Linguistics, 1982, 8, 55-83.

Joshi, A., Webber, B., & Sag, I. (Eds.). Elements of Discourse Understanding. Cambridge, England: Cambridge University Press, 1981.

Reitman, W. (Ed.). Artificial Intelligence Applications for Business. Norwood, NJ: Ablex Publishing Corp., 1984.

Schank, R.C. & Riesbeck, C.K. (Eds.). Inside Computer Understanding. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1981.

Sparck-Jones, K. & Wilks, Y. (Eds.). Automatic Natural Language Parsing. New York: Halsted Press, 1983.

Winograd, T. Language as a Cognitive Process, Volume 1, Syntax. Reading, MA: Addison-Wesley Publishing Co., 1983.

Woods, W. Natural language communication with machines: an ongoing goal. In W. Reitman (Ed.), Artificial Intelligence Applications for Business. Norwood, NJ: Ablex Publishing Corp., 1984.

### 2.3.4 Summary

Natural language understanding research is already yielding commercial products based on results in syntactic processing and semantics in narrowly, precisely defined domains, such as access to a single data base.

Both natural language understanding and natural language generation offer great potential not only in making computers more usable to those who are not professional programmers but also in making new computer applications. These potential applications include:

- o providing briefings and explanations appropriate to the background of the individual
- o communicating with expert systems that understand the person's intention, whether the language he/she uses is polished or ill-formed
- o appropriately responding to the person's intention and need, for instance, in requesting help, explanation, or performance of a task
- o providing intelligent coaching for someone learning a new language

Nevertheless, there is much research required to achieve that potential. Syntax, the study of how words and phrases are combined to make meaningful expressions, still needs some study, particularly in providing a unified treatment of understanding and generation, in understanding ungrammatical forms, and in employing the nuances of particular words and syntactic constructions. In semantics, research in knowledge representation and in formally representing vague terms such as "few" and "very" is needed. Semantics is less well understood than syntax, but is more advanced than pragmatics, the study of the influence of linguistic context, beliefs, goals, and the situation on the meaning and intention of communication. In pragmatics, modeling contextual factors and their impact on the meaning of expressions requires much research in order to achieve natural, helpful communication.

Additionally, substantial breakthroughs are needed so that the underlying application need not be so constrained. Systems that can communicate about many overlapping domains (e.g., overlapping data bases or overlapping expert systems) are many years away, though certainly feasible in the future.

## 2.4 KNOWLEDGE REPRESENTATION

### 2.4.1 Overview

Researchers in knowledge representation (KR) take as their primary goal the development of techniques to allow information about the world to be stored in a computer so that this information may later be used and new information inferred. The author of any computer program, no matter how small or large, must make choices regarding representation. Even a trivial program that, for example, calculates amounts of money must represent these amounts in some manner, where there are typically a variety of ways to do so. But there is a distinction between representation in general and knowledge representation in particular. While no clear line of demarcation can be drawn around the field of knowledge representation, it has several distinguishing characteristics. First, the long term goal is to develop a computer language that has the expressive power of a natural language, such as English. Second, its goal is not only to store and retrieve this information but to infer all new information that is logically deducible from it. A third goal that is increasing in importance is the ability to reason, which goes beyond inferring that which is logically deducible to that which is plausibly inferable. If one drops the requirement of inferring and reasoning, then the task is greatly simplified, and one finds oneself closer to the field of data bases rather than knowledge representation.

As an example, let us examine the representational needs of a computer system that attempts to understand English; in particular the system (Kernel) being developed at BBN under DARPA support is a model. The system has several components, the first of which parses a sentence that is typed by a user. The parsed sentence is then translated into a literal interpretation. Following that, a third component applies pragmatics to the sentence, after which the full meaning of the sentence is determined. Take the sentence "Can you pass the salt?". After being parsed, the literal interpretation describes this as a yes/no question that asks whether the listener has the ability to pick up and move some quantity of salt. Already, we need to represent objects such as salt and the parts of the robot that could pick it up and move it, such as its arm, and we must represent the notion of the ability of the robot to use its arm to pass the salt. But the literal interpretation is often not what is really intended. Rather than a yes/no question, the speaker is probably asking for the listener to actually pass the salt and not to answer yes or no. Now we need to represent the intentions of the speaker and we need to represent his or her needs in the current context. Furthermore, we need to reason that his or her needs at, say, a dinner table lead us to believe that the speaker actually wants the salt passed, though in other contexts, the meaning may be different.

Inferring and reasoning are generally considered within the realm of intelligent behavior, and as a result, knowledge representation (KR) is a concern of all researchers in artificial intelligence. It is not that all AI researchers focus on KR, but each must address it at some point in his or her work. Of the many groups that do focus on KR, nearly all do so in the context of other research interests, such as natural language understanding or computer vision, and these groups concentrate on the particular knowledge representation problems that arise in their related projects. In fact, since it is impossible to represent "everything," topics in representation must be selected with some type of application in mind. This produces a spectrum in KR research that ranges from techniques that are applicable across a wide range of applications to those that specialize in just one.

Usually, each KR group designs and/or constructs a computer program that embodies its ideas, and each such program has:

- a description language for specifying information
- mechanisms for retrieving and inferring information

Regarding examples of description languages, "Member(Clyde,Elephants)" might be a way of stating that Clyde is a member of the set of all elephants, which is an indirect way of saying that Clyde is an elephant. "Subsumes(Mammals,Elephants)" might be a way of stating that all elephants are mammals. Regarding examples of query languages, "Member(X,Elephants)?" might be a way of asking "Which elephants do you know about?". Here, X is a variable and the query processor might return with X bound to the set of all elephants that have been mentioned via the description language. Alternatively, we might pose the query "Member(Clyde,Elephants)?" as a question that should be answered either yes or no. But what of the query "Member(Clyde,Mammals)?"? Given the above statements that Clyde is an elephant and that all elephants are mammals, the program should be able to infer that the answer is "true," and indeed, most KR programs would be able to do so.

Some KR systems also allow one to describe relations that are typically, but not always, true. In such systems, for example, we could state that elephants are typically gray. Usually, these systems have a mechanism for drawing plausible inferences from such statements, where the mechanism is based upon the idea that if there is no evidence to the contrary, assume to be actually true that which is typically true. So, if we asked for the color of Clyde the elephant and if there was no evidence to the contrary, the system would plausibly infer that Clyde's color was gray. Of course, the system might later be informed that Clyde was an albino elephant, leading the system to retract the statement of his color being gray and to retract any conclusions that were reached based on Clyde's being gray. This type of reasoning is also called default reasoning; here, our default is that in absence of contrary evidence, an elephant is colored gray.

A semantics for a language is an account of what the sentences in the language mean with respect a given domain. A semantics for the language of the examples from this section would guide us in determining the precise meaning of "Member(Clyde,Elephants)" and "Subsumes(Mammals, Elephants)", and furthermore, would explain why the two taken together lead to the conclusion that "Member(Clyde,Mammals)". Unfortunately, most KR researchers are lax in formally specifying a semantics for their representation languages, and instead are quite informal, leaving the operational semantics of a KR computer system to be the final arbiter. Thus, users of such programs may need to guess or to discover by trial and error certain subtle questions of meaning.

These points immediately raise some crucial questions regarding KR systems. Let us assume that one has in mind a particular domain and class of problems, and that she or he is evaluating a particular KR system. Since each description language is limited in its expressive power, to what extent can the description language satisfactorily capture the relevant information from the domain? A similar question should be asked regarding the mechanisms for retrieval, i.e., can all information that is stored be retrieved readily? But more important is the extent to which the system can infer new information and the manner by which inferences are made. Can the system make the inferences that the given problem requires? Will the inference mechanism work quickly enough? Will it avoid making lots of inferences that are not of use? If plausible inferences are needed, are the necessary mechanisms available?

Unfortunately, this approach using yes/no questions is somewhat misleading as the problems of representing and inferring knowledge are far more complex than it suggests. Rather, the above should be construed as dimensions for evaluations to be made. It is unlikely that well-tailored fits can be readily made between the needs of an application and the properties of an existing KR system. This is due, at least in part, to the extremely sensitive balance between that which can be expressed versus that which can be inferred in a reasonable amount of computer time. For applications, one must avoid combinatorial explosion--a problem suffers from combinatorial explosion if it requires so many steps to solve that it is simply not solvable given any reasonable amount of resources. Of course, one wants a KR system in which one can state just about anything and to be able to infer likewise. Unfortunately, it is all too easy to design a system that infers so much that it is impossible to control. While searching for a way to infer a certain fact, it follows many, many blind alleys. Continuing with our earlier example, when attempting to answer the question "Member(Clyde,Mammals)?", the crucial fact to use is "Subsumes(Elephants,Mammals)". But the system may have hundreds or thousands of other facts about Clyde in particular, or elephants in general, and the system may not have information that tells it which information is relevant to the question at hand. Thus, it could try to use the other thousands of facts first in attempting to answer the question. Even worse, it is very easy to design systems that cannot guarantee that the questions one might ask are even decidable.

### 2.4.2 Glossary

Decidable: A problem is decidable if it can be viewed as a yes-no question, and a computer program can be written which is guaranteed to halt in a finite amount of time given any instance of the problem and to correctly answer yes or no.

Exponential time: Suppose the size of an input can be measured as the integer n. An algorithm is said to run in exponential time if it would take computer time on the order of $2^n$ on inputs of size n, $n > 0$.

Expressive power: The expressive power of a KR language is the class of statements that can be made in that language.

First-order predicate calculus: A class of languages developed in mathematical logic that are used by some KR systems.

Frame language: A KR language where information is organized around units in a hierarchy.

Horn clause: A logic statement of the form:

$$A_1 \text{ and } A_2 \text{ and } \ldots \text{and } A_n \text{ implies } C$$

where C and each Ai are simple assertions. All programs in PROLOG are written in this form.

Inferential closure: The set of statements deducible from all possible, valid inferences no matter how long the chain of reasoning steps, given a set of axioms and a set of rules for drawing valid inferences.

Inferential tractability: The property that any valid inference can be drawn in polynomial time, given the length of the conclusion.

Inference: A conclusion or the process of drawing conclusions.

Inheritance: The property in a hierarchy that a lower frame has associated with it (by inheritance) all the information associated with all of its ancestors in the hierarchy.

Knowledge representation: A computational means of formally representing information, which would be called knowledge in a human.

Logical deduction: In logic, the means of drawing valid inferences given a set of axioms and a set of inference rules.

Logical representation language: A KR language based upon a mathematical logical language.

Plausible inference: An inference which is reasonable but may not be valid logically.

Polynomial time: An algorithm is said to run in polynomial time if for any input of size n, n > 0, the algorithm computes the answer using time that is a polynomial in n.

Resolution theorem proving: A particular means of doing logical deduction in first-order predicate calculus. Only one inference rule ("resolution") is used, and all formulas have been converted to a standard ("normal") form.

<u>Semantic network language</u>: A class of KR languages based on labeled, directed graphs of mathematical graph theory.

<u>Subsumption</u>: A particular relation between formulas in a logic or between sets. A formula B subsumes a formula A if whenever A is true, B must also be true. In a similar way, a set B subsumes A if A is a subset of B.

### 2.4.3 State-of-the-Art

We first describe the broad categories of work in KR, followed by the survey information.

Styles of description languages for KR systems fall into 3 general categories; logic languages, semantic networks, and frame languages.

#### Logic Languages

The name "logic languages" is misleading as it implies that other languages are not logical, which is not the case. The intent of the category name is to show that these languages have a nearly one-to-one correspondence to some language from mathematical logic, the most popular ones being first order predicate calculus (FOPC) and a well known subset of FOPC, Horn clauses. The primary advantages of these languages from logic are that they have (1) a wide expressibility, (2) a formally specified semantics, and (3) a general mechanism for inference. These languages provide a good example of the trade-off between expressibility, inferential capability and inferential tractability. FOPC has more expressive power than Horn clauses, as the latter is a subset of the former. For FOPC, resolution theorem proving is a technique that will infer all that is logically deducible from a given set of sentences (i.e., information). But for FOPC, resolution is semi-decidable--

i.e., some attempts at proving a sentence that is in fact invalid can theoretically take forever. Consequently, one usually imposes a resource limit; if those resources run out, the program returns with "don't know." Since Horn clauses constitute a smaller language, resolution theorem proving is more tractable. A proof cannot theoretically take forever. In fact, it can take exponential time at most and, by restricting the language still further, polynomial time. Thus, one must carefully weigh one's representational and inferential needs when choosing among these, and indeed all, KR languages.

## Semantic Networks

The second category is that of semantic networks. A semantic network is composed of nodes and links, each link connecting a pair of nodes. A node can be named, but is otherwise without structure, and usually represents either an object or a set of objects. A link can be named, is without structure, and represents a relation between either objects or sets of objects. Semantic networks offer a wide expressibility although typically without a clear semantics. The claim is that semantic networks simplify the search for information relevant to a given entity because the links between nodes are directly accessible from each connecting node--i.e., the information about an object is "physically close" to the node representing the object. However, in practice, this claim has never been clearly shown to be true. An important relation between nodes in almost all semantic network systems is that of subsumption, sometimes called "ISA." Like all relations, subsumption is represented by a link between nodes. Usually such nodes represent sets and the subsumption link means that the subsuming set includes the subsumed set--subsumption is like set inclusion. It is an important relation because it appears so often. To say that all elephants are mammals, one adds a subsumption link from the node for elephants to that for mammals. Several types of inference have been found useful with semantic networks, particularly inheritance. Inheritance works between nodes with subsumption links connecting them, and it enforces the notion that properties of the members of a set are also properties of members of subsets of the set. In other words, if mammals are warm blooded and elephants are mammals, then elephants are warm blooded. Nearly all semantic network systems perform inheritance automatically, and some perform other types of specialized inference. This contrasts with the logical languages in that researchers of semantic network systems have concentrated upon various types of specialized inference and have not attempted mechanisms for inference in general.

### Frame Languages

The third category is that of frame languages. Here, the primary unit is a frame that, like nodes in semantic networks, usually represents an object or set of objects. A frame has a name and a collection of slots. Each slot is named, represents a relation and has an associated filler. This is similar to semantic networks, except that the fillers of a slot need not be other frames (for example, they could be procedures), and furthermore, each slot of a frame can have additional information stored with it. Thus, a wide variety of information can be captured. Regarding inference, frame and semantic network languages are similar--researchers for both have provided specialized inference mechanisms, inheritance in particular, but not general inference mechanisms.

### 2.4.3.1 Operational Application

There are no operational applications in KR; there are only demonstrable systems that form components of expert systems, natural language processors, etc. These serve as the knowledge base or as the data base of an application.

### 2.4.3.2 Principal Areas of Research

Several current, well-known KR systems fall into a new category called hybrid KR systems. In these, a KR system is viewed as having two or more components, where each specializes in what it can represent and the types of inferences it can perform. The hope is that by "carving up" one's representational and inferential needs into efficient components, one can hope to get wide expressibility with an efficient inference capability. Of course, the problem is in the "carving up" and in the system's ability to transfer information between components. This is a promising outlook being explored at BBN with the KL-TWO system, and at Fairchild with KRYPTON. Both of these systems include a component for describing terms based on earlier semantic network and frame languages, and a second component for making assertions about the world using those terms.

Regarding the logical languages, there is much work using the PROLOG programming language that is akin to KR, although in general, PROLOG belongs under the heading of programming tools. In general, users of PROLOG first write a KR system in PROLOG and then use that KR system as if it were written in any other programming language, for example, WARPLAN (Warren, 1976). Many AI researchers who use FOPC as their representation language simply assume that a resolution theorem prover will be able to supply their inferential needs. At the current time, this is an incomplete strategy as a theorem prover is far from a simple tool. But work on theorem proving continues and looks promising, making it a reasonable long term bet.

Regarding frames, the UNITS system is the most well known current work. It embodies the ideas discussed earlier and includes many tools for aiding one who is building a knowledge base. The UNITS package is now a component of the KEE system commercially available from IntelliCorp. Also, the predecessor to the KL-TWO system developed at BBN, KL-ONE, was a KR system based largely on the ideas of frames and, to some extent, semantic networks. KL-ONE has been superseded by KL-TWO.

The probability of success in these research areas is .9, where "success" here means incorporating the resulting knowledge representation ideas into operational expert systems or natural language processors.

#### 2.4.3.3 Major Gaps and Problems

All of these types of approaches have reached a level of maturity such that languages have been used in commercial or prototype expert systems or in prototype natural language understanding systems. In another sense, it is clear that none has yet achieved the level of expressibility or the level of inference support that their creators dream of. This is elaborated in the next sections.

Much of the other well known work in KR is dedicated to particular types of problems, each of which still presents enormous difficulties to AI. Briefly stated, these are the representation of defaults (or typicality) information, actions and events, space, time, mutable objects, and propositional attitudes (e.g., beliefs and wants). Additionally, drawing analogies based on representations is another gap. Each of these problems is important for the development of more sophisticated systems.

The problem of representing defaults deals with an essential component of human reasoning. One often needs to:

- make decisions based on what is normal
- justify a decision
- recognize what conclusions should be retracted in light of previous assumptions proving inappropriate

The type of reasoning this typifies is called nonmonotonic reasoning.

In each of the classes of KR languages discussed earlier, research has begun on this problem. Since it is so fundamental, it may be very long before fully adequate solutions are found. Partial solutions should contribute significantly to applications as the work progresses.

Representations of actions, events, space, time, and mutable objects are all interrelated. This may be obvious for the first four since actions can result in events, and both obviously occur in space and time. Actions and events effect objects by possibly imposing change upon them, as in the event of an explosion reducing a small building to rubble or in the action of wandering through a snow covered landscape causing snow blindness. The problem in all of these is to represent common sense knowledge and common sense reasoning.

Both this problem and nonmonotonic reasoning have proven to be critical for future generations of expert systems and natural language processors. The basis for this conclusion is that not all facts necessary for decision-making can be reduced to numbers, systems of differential equations, etc. For instance, in the example of the explosion, the appropriate conclusion for a robot might be to duck to avoid flying debris. Even if one could reduce certain knowledge to numbers, trajectories, and equations, it may be more expedient to simply represent it symbolically as in the case of the robot's need to duck flying debris. Other knowledge is simply vague or incomplete. For example, "Few enemy X aircraft are equipped with jamming facility for transmissions such that... ."

Representation of beliefs and desires is also critical, because of several needs:

- the need to predict the beliefs and knowledge of colleagues and adversaries in order to appropriately assess, plan, etc.
- the need of some expert systems to reason about likely adversarial action
- the need of natural language understanders to interpret input in terms of beliefs and wants (e.g., so that "Can you predict its ETA?" is interpreted as a command rather than a yes/no question)
- the need of natural language generators to communicate effectively given the expertise of the listener

This is a significant problem for reasoning because even when we know that A believes "X" and that A believes "if X then Y," we do not know whether A believes "Y." If we did, mathematicians would not have to struggle to discover theorems, scientists would not have to work to know the consequences of a theory, and other experts wouldn't have a problem in knowing the implications of a new datum. Reasoning by analogy is also critical to problems such as situation assessment and advising, for the analogy may suggest a general framework of solution while the differences from the analogous can imply concrete aspects needing attention. It provides ways of viewing one thing differently, an important aspect of creative intelligence.

Consequently, representing and reasoning about beliefs and desires are a fundamental unsolved problem.

All of the problem areas discussed are being pursued.

#### 2.4.3.4 Approaches That Failed

As in other cases, it is not so much that approaches have failed as that they have evolved, in fact coming close together. Consequently, what is clear is what characteristics an approach should have:

- o at least the expressive power of Horn clauses of first-order predicate calculus (FOPC) and perhaps more than even a first-order language
- o a reasoning capability that is computationally tractable, as opposed to solely a complete theorem proven for FOPC
- o nonmonotic reasoning
- o adequate representation of actions, events, space, time, mutable objects, beliefs, and wants
- o the ability to draw analogies

No approach is near attaining all of these desiderata.

#### 2.4.3.5 Major Laboratories and Key Contact Points

- o BBN, Cambridge, MA/Rusty Bobrow, David Israel, Jim Schmolze
- o Carnegie-Mellon University, Pittsburgh, PA/Allen Newell, Scott

Frahlman

- o Fairchild, Palo Alto, CA/Ron Brachman
- o IntelliCorp, Palo Alto, CA/Richard Fikes
- o MIT, Cambridge, MA/Gerald Sussman
- o Rand Corporation, Santa Monica, CA/Henry Sowizral
- o SRI International, Menlo Park,CA/Robert Moore, Jerry Hobbs
- o SUNY Buffalo, Buffalo, NY/Stuart Shapiro
- o Stanford University, Stanford, CA/John McCarthy
- o University of Rochester, Rochester, NY/James Allen
- o University of Toronto, Toronto, Canada/Hector Levesque, John Mylapolous
- o USC/Information Sciences Institute, Marina del Rey, CA/Bill Mark
- o Xerox PARC, Palo Alto, CA/Daniel J. Bobrow

### 2.4.3.6 Recommended Key References

Bobrow, D.G. (Ed.). Artificial Intelligence: Special Issue on Non-Monotonic Logic, 1980, 13.

Bobrow, D.G., & Collins, A. (Eds.). Representation and Understanding. New York: Academic Press, Inc., 1975.

Bobrow, D.G., & Winograd, T. An overview of KRL, a knowledge representation language. Cognitive Science, 1977, 1, 3-46.

Brachman, R.J., Fikes, R.E., & Levesque, H.J. KRYPTON: A functional approach to knowledge representation. IEEE Computer: Special Issue on Knowledge Representation, 1983, 16, 67-73.

Brachman, R.J., & Levesque, H.J. The tractability of subsumption in frame-based description language. Proceedings of AAAI-84, Austin, TX, 1984, pp. 34-37.

Brachman, R.J., & Schmolze, J.G. An overview of the KL-ONE knowledge representation system. Cognitive Science, forthcoming.

Brodie, L., Mylopoulos, J., & Schmidt, J.W. (Eds.). On Conceptual Modeling. New York: Springer-Verlag, 1984.

Carbonell, J. Derivational analogy and its role in problem-solving. Proceedings of AAAI-83, Washington, D.C., 1983, pp. 64-69.

Findler, V. (Ed.). Associative Networks. New York: Academic Press, 1979.

Hayes, P.J. In defense of logic. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Pittsburgh, PA, 1977, pp. 559-565.

Hayes, P.J. The logic of frames. In D. Metzing (Ed.), Frame Conceptions and Text Understanding. Berlin: Walterde Gruyter, 1980.

Levesque, H.J. A fundamental tradeoff in knowledge representation and reasoning. Proceedings of CSCSI-84, London, Ontario, 1984, pp. 141-152.

Levesque, H.J. A logic of implicit and explicit belief. Proceedings of AAAI-84, Austin, TX, 1984, pp. 198-202.

Rich, E. Default reasoning as likelihood reasoning. Proceedings of AAAI-83, Washington, D.C., 1983, pp. 348-351.

Warren, D.H.D. Generating conditional plans and programs. Proceedings of the AISB Summer Conference, Edinburgh, Scotland, 1976, pp. 344-354.

Woods, W.A. What's in a link: foundations for semantic networks. In D.G. Bobrow & A. Collins (Eds.), Representation and Understanding. New York: Academic Press, Inc., 1975.

### 2.4.4 Summary

Knowledge representation is the task of providing a formal representation (i.e., one whose syntax and semantics is well-defined) for knowledge such as facts, plans, rules of thumb, etc., in a way that supports reasoning based on that knowledge. Consequently, it is at the heart of AI research and applications.

Though there are three classes of knowledge representation languages, there is general agreement about the goals of such a language:

- o to be able to represent any piece of knowledge expressible in English
- o to support common sense reasoning and reasoning about another's beliefs and knowledge
- o to support reasoning and decision-making performed in real-time

While knowledge representation has advanced sufficiently to support laboratory prototypes and even commercial products in expert systems and natural language processing, there are a number of fundamental gaps that need research. These include:

- o representation of actions, events, space, time, mutable objects, beliefs and desires
- o support for common sense reasoning
- o support for determining how much time to focus on and reason about a given hypothesis or goal

Since applications of expert systems and natural language processing depends so centrally on knowledge representation, this is an area that is critical for progress in AI.

## 2.5 COMPUTER VISION (IMAGE UNDERSTANDING)

### 2.5.1 Overview

Image Understanding is the automatic mapping between regions of an image or images, and descriptions of those regions. The descriptions of the image regions will normally refer to nodes in a knowledge representation system, and these nodes will normally correspond to objects in the imaged domain.

Computer vision offers the potential of gathering information at long range or even at the microscopic level, depending on the sensor and aids to it. Furthermore, this may be possible with frequencies well beyond the (human) visible spectrum.

Machine vision is still at an extremely restricted and early stage. However, partly as a result of increasing understanding of the mechanisms of human vision, and partly from efforts to develop algorithms and systems for limited cases, machine vision system are gradually growing in capabilities.

Special-purpose systems exist for inspection of industrial parts, analysis of aerial photographs, and processing of chest x-rays. Current computer vision systems are based on domain-specific constraints and techniques, in contrast with such areas as natural language processing, where some system components are very general-purpose and domain independent. Therefore, at present only special-purpose tasks may be off-loaded from a human.

2.5.2 Glossary

2 1/2 D: A representation of the depth and orientation of the observable surfaces of an image, including abrupt changes in depth or orientation.

Gray level: The intensity of light at a pixel in an image resulting from a sensor, such as a black and white TV camera. Usually, an integer between 0 and a power of 2, e.g., 0-63.

Image preprocessing: The earliest processing of image data, such as removing sensor distortion, and standard patterns of noise.

Feature extraction: Identifying features (such as blobs and edges) in an image. Patterns of features are used to recognize texture and shading change, contour, etc., to determine some three-dimensional information.

Pixel: A point in an image.

Region growing: Determining which pixels form a region in the image, based on homogeneous properties. Regions have size and shape, for instance, as symbolic properties, rather than representation purely as icons.

Object modeling: Representing important characteristics of an object for purposes of identifying a collection of regions as an instance of the objects.

### 2.5.3 State-of-the-Art

#### 2.5.3.1 Operational Applications

The potential applications of image understanding (IU) are very broad. Because IU is computationally demanding, most applications have required substantial software, firmware, and hardware customization. Among the applications that are current and near-term are medical image analysis, robotics, parts inspection, and a host of everyday applications like traffic light control. These applications range greatly in difficulty, and many applications can be accomplished with comparatively *ad hoc* methods; other applications, such as terrestrial navigation, require extensive knowledge of the objects in their domain, and powerful methods for segmenting the image and identifying the objects.

#### 2.5.3.2 Techniques that Make for Effective Operational Systems

Effective Control of Illumination.

For example, the use of structural light changes the computationally difficult stereoscopic depth determination into a trivial trigonometry problem. In other circumstances, either multiple light sources of different colors, different frames taken with different illumination or moving light sources, can provide significant assistance in the analysis of shape. In other cases, the control of the illuminating spectrum and the spectral sensitivities of the imaging device can take advantage of the inherent characteristics of the imaged object to improve, contrast, or otherwise simplify the subsequent processing and improve the overall reliability of the understanding process. In other words, when possible, tailor the illumination to the problem.

### Type and Placement of Imaging Devices

Many of the computational problems can be simplified by selecting imaging devices which provide all and only that image information that the understanding system requires. In the scene to be understood, the number and placement of cameras or other imaging devices can eliminate many image understanding problems and simplify others. For example, problems of foreground objects occluding background features of interest, and problems of difficult edge detection due to juxtaposed textures can be eliminated by appropriate camera angle. In many circumstances, the use of multiple cameras enables the dynamic selection of views to minimize these difficulties. Many powerful algorithms for understanding three-dimensional objects are based on the reconstruction of the three-dimensional shape from multiple views.

### Spatial Resolution

Since most image processing and image understanding applications require image information only up to some limiting spatial frequency, and because the amount of image data to be processed increases as the square of the limiting spatial frequency required by the application, significant advantages can be gained for some applications by matching anisotropic or nonuniform spatial sensitivity in the sensors to similarly and anisotropic, or inhomogeneous application needs. Often significant application advantages can be gained by the careful selection of sensor spectral bands for the sensors. Linearly or otherwise combining images from different spectral bands often brings out image characteristics which greatly simplify the understanding process. In many applications the spatial nonuniformities in the imaging device must be corrected through spatially dependent calibration of the imaging device; in other words stability of the imaging device and illumination is important in these applications.

Processing steps (defined in the glossary) for efficient operating systems include:

- o image preprocessing
- o feature extraction
- o region growing
- o object modeling

### 2.5.3.3 Principal Areas of Research

Connectionist Models of Vision

This area of research is investigating a new model of parallel computation. In the connectionist paradigm, a UNIT is a computational entity which is a measure of the activation of this unit with up to about one thousand inputs which are measures of the activations of other units and one thousand outputs. The basic computation is the derivation of the activation of each unit based on the inputs and the predetermined influence. Connectionist computations are performed iteratively, with the activation values of the input units affecting the output activations in the immediately succeeding cycle.

In connectionist models of vision, the input activations for the first level of units are provided directly by the intensity of other values at specific image sensor receptor sites. The first level may model simple low-level features such as edges, the second level may model higher level spatial features or motion in the edges, and higher levels may model features such as orientation, position, and objects.

This research is in its very early stages. No really adequate processing resources are available at this time; the complexity of the models which can be tested is still quite low. Nonetheless, these very simple models, involving typically fewer than one thousand units, have exhibited robust feature recognition and even a simple form of learning or adaptation to optimize feature discrimination.

At this stage in the development it is difficult to determine if the technology will have any limitations; all that can be said is that these systems show promise for performing at human or super human levels, with no clear limitations that have been identified.

## Conventional Computational Paradigms

The other principal research areas assume conventional computer architectures, rather than the connectionist model. Since this section assumes that framework, these may be described more succinctly than the connectionist model. Work on feature extraction continues, for instance, by identifying regions which might have arisen from similar process (Pentland, 1984), by employing the information in multiple resolution (Witkin, 1983), and by extracting shape information. At higher levels of image processing, work is progressing on representing more complex shapes; see Brady (1983), Barr (1981) and Pentland (1984). Strategies for indexing into large data bases are also beginning to be investigated for object recognition.

All of these are incremental improvements on the techniques previously outlined in this section. Though the details of those research topics are relatively new, they are being explored within a rather mature paradigm and probably have a .8 probability of yielding successful applications therefore. Limitations of the current state-of-the-art are described in the following section on gaps.

### 2.5.3.4 Major Gaps and Problems

No adequate theoretical AI models yet exist of space and spatial relationships. See the section on Knowledge Representation. Representation for the shapes and visual appearance of objects is another active topic in knowledge representation.

No really capable image gathering devices with ability approaching that of the retina have been produced. Sensor research is at least as much electrical engineering, mechanical engineering, etc., as AI; nevertheless, the quality of the sensor obviously impacts the quality of results.

No systems or architectures have yet been produced which remotely approach human performance in robustly generating object descriptions from their manifestations in the image. Success has been achieved where the problem has been simplified, using the techniques described earlier for effective operational applications.

### 2.5.3.5 Approaches that Failed

The only approach that one can say in some sense failed is that of the perceptron; interest tapered off after analytical analysis of its potential appeared in Minsky and Papert (1969). Yet, even in this case, the goal of neural modeling and basing the computations on numerical "activation levels" has resurfaced in the connectionist paradigm now that more is understood about the architecture of vision components. Now that there is basic agreement on framework of correcting for sensor distortion, extracting features, growing regions, and recognizing objects, it makes sense to explore how one might achieve algorithms for these components in computational units closer to neurons than to conventional computers.

### 2.5.3.6 Major Laboratories and Key Contact Points

- o Carnegie-Mellon University, Pittsburgh, PA/Raj Reddy
- o Fairchild, Palo Alto, CA/H.G. Barrow, J.M. Tenenbaum, A. Witkin
- o Machine Intelligence, Mountain View, CA/Charles Rosen
- o MIT, Cambridge, MA/Berthold Horn, Patrick Winston
- o SRI International, Palo Alto, CA/Robert Bolles, Martin Fischler, Alex Pentland
- o Stanford University, Stanford, CA/Thomas Binford
- o University of Maryland, College Park, MD/Azriel Rosenfeld
- o University of Pennsylvania, Philadelphia, PA/Ruzena Bajscy
- o University of Southern California, Los Angeles, CA/Ramakant Nevatia
- o University of Rochester, Rochester, NY/Dana Ballard, Jerome Feldman

### 2.5.3.7 Recommended Key References


Agin, G.J. Computer vision systems for industrial inspection and assembly. _Computer_, 1983, _13_, 11-20.

Ballard, D.H., & Brown, C.M. _Computer Vision_. Englewood Cliffs, NJ: Prentice-Hall, 1982.

Barr, A.H. Superquadrics and angle-preserving transformation. _IEEE Computer Graphics and Applications_, 1981, _1_, 11-23.

Barrow, H.G., & Tenenbaum, J.M. Computational approaches to vision. In K. Boff, L. Kaufman, & J. Thomas (Eds.), _Handbook of Perception and Human Performance_. New York: John Wiley & Sons, 1985 (in press).

Brady, M. Criteria for representations of shape. In J. Beck, B. Hope, & A. Rosenfeld (Eds.), _Human and Machine Vision_. New York, NY: Academic Press, 1983.

Hanson, A.R. & Riseman, E.M. _Computer Vision Systems_. New York: Academic Press, 1978.

Marr, D. *Vision*. San Francisco: W. H. Freeman, 1982.

Minsky, M.L. & Papert, S. *Perceptions: An Introduction to Computational Geometry*. Cambridge, MA: The MIT Press, 1969.

Pentland, A.P. Shading into texture. *Proceedings of AAAI-84*, Austin, TX, 1984, pp. 269-273.

Pratt, W.K. *Digital Image Processing*. New York: Wiley-Interscience, 1978.

Rosenfeld, A. (Ed.). *Human and Machine Vision*. New York: Academic Press, 1983.

Ullman, S. *The Interpretation of Visual Motion*. Cambridge, Massachusetts: The MIT Press, 1979.

Ullman, S. (Ed.). *Image Understanding 1983: Advances in Computational Vision*. New Jersey: Ablex, 1984.

Witkin, A.P. Scale-space filtering. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Los Altos, CA: William Kaufmann Inc., 1983, pp. 1019-1023.

### 2.5.4 Summary

The potential of computer vision is great, both in everyday and in military applications. Though replicating human recognition capabilities in natural scenes is not feasible in the foreseeable future, there are several simplifying assumptions that lead to operational applications, such as control of illumination, placement of sensors, and matching spatial and color resolution to application needs.

The standard components of computer vision systems include:

- o digitization to convert the analog input signal to a matrix of numbers, e.g., representing the intensity in a small portion of the image

- image preprocessing, e.g., to remove sensor distortion
- feature extraction, the process of hypothesizing edges
- postulating higher level properties such as regions
- identifying objects as related regions

One of the most critical needs is more powerful computing, since one is generally dealing with relatively large matrices of numbers, e.g., 512 X 512. Other major gaps include representing spatial information about objects and improved sensors.

## 2.6 TUTORING AND TRAINING

### 2.6.1 Overview

Intelligent tutoring and training systems are those which contain some measure of expertise which is useful in the educational process.

One class of such programs is closely related to other "Expert systems"; they include extensive knowledge of an expert domain. As such, they have the potential (sometimes as yet unrealized) to be used as educational tools. Some of the programs which have possibilities for training are INTERNIST (Pople, 1977) and MYCIN (Shortliffe, 1976), both medical expert systems; PROSPECTOR (Duda, Gaschnig & Hart, 1979), a system which gives advice on possible mineral deposits; MOLGEN (Friedland, 1979), an expert system in designing experiments in molecular genetics; DENDRAL (Lindsay et al., 1980), an expert chemistry system used in determining the molecular structure of unknown organic compounds; and STEAMER (Stevens, et al., 1981), a knowledge-based simulation of a steam plant used to power Navy ships. We will explore here the potential of this class of programs by describing STEAMER in some detail.

STEAMER is a sophisticated computer system that creates a small, portable, and inexpensive version of a steam propulsion plant. It contains a mathematical model of the propulsion plant, dynamic graphic displays of different views of the plant (e.g., gauge, pumps, valves), a graphics editor for creating new views, and the capability for setting up mini lab learning contexts in a variety of science and engineering areas. STEAMER runs on a small but powerful minicomputer, a Symbolics 3600 using two screens for output (one for the color graphics display, the other for typing commands), and a combination of the keyboard and a mouse for input.

A black and white print of a typical STEAMER screen is shown in Figure 2. The band above the italicized word "Commands" shows 18 commands that may be selected by using a pointing device (mouse). Some status information is printed below that using a mixture of English descriptions and readings of certain measurements, such as speed, RPM's, and drum pressure. The lower left square allows typing Lisp commands. Part of a schematic, including guages, is displayed at the lower right.

A steam plant trainee using STEAMER chooses one view of the plant (such as the throttle gauges), sets some system parameters, and runs the plant, watching how the gauges change. The trainee can then choose to view the internal operation of the plant (e.g., the main engine gland seal) and watch the flow of steam as gauge-driven valves open and close. A trainee can simulate a "casualty" (e.g., a valve failing and being stuck in the open position), and look at its propagation through the system by selecting other views of the plant. Creating and following such a catastrophe would of course be impossible in the actual steam plant.

STEAMER also contains the facilities for trainers to create mini labs so that trainees can study specific topics pertinent to the steam plant domain. By using the graphics editor, a trainer can create a lab which demonstrates the relationship among four different kinds of pressure gauges. In such a lab, each gauge will change when any one changes, visually reflecting the conversion formulas. This mini lab facility can be used to teach such

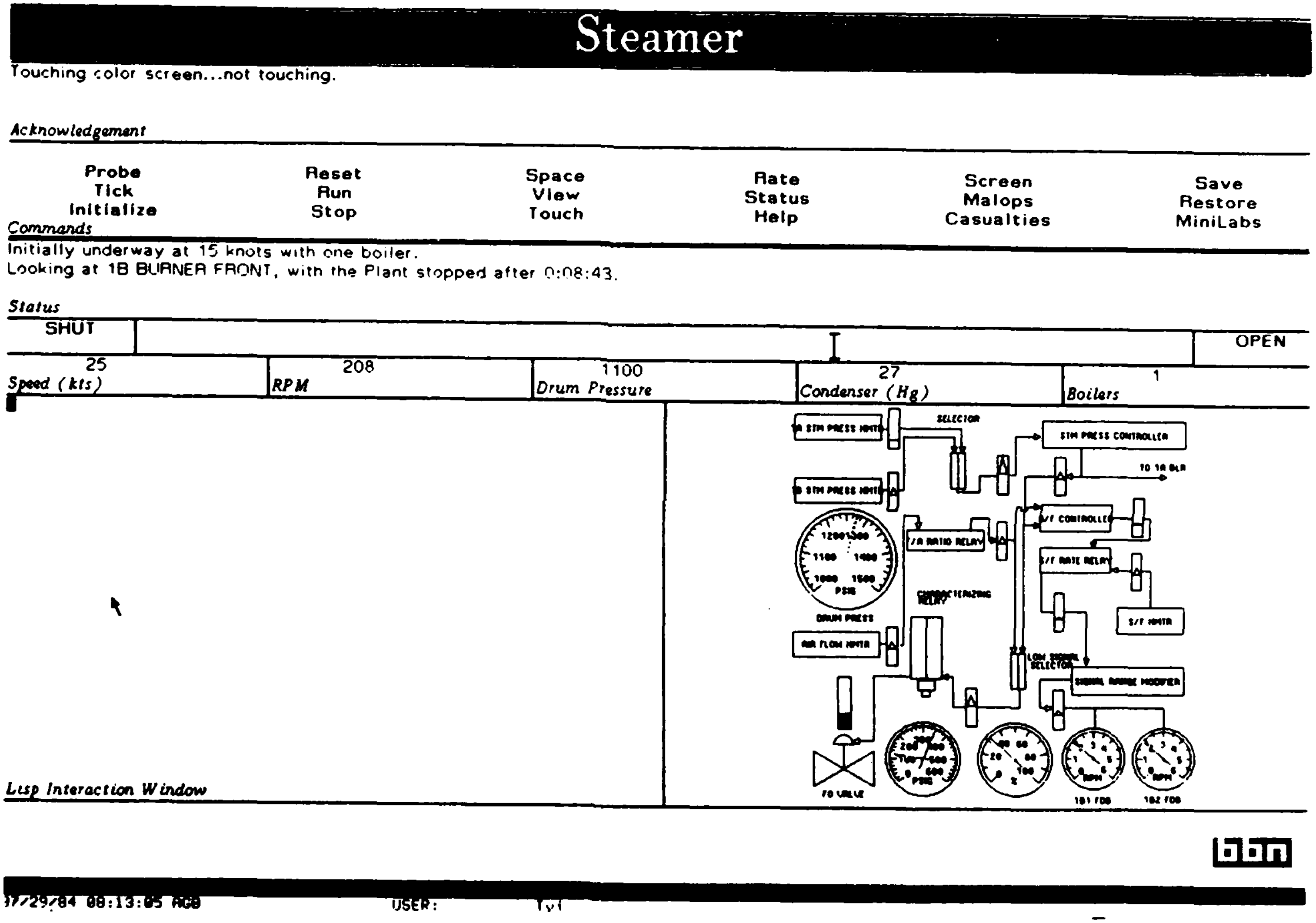



Figure 2. A STEAMER DISPLAY
(Description of display is in text.)

traditional science curriculum topics as Fahrenheit to Celsius conversion, the relationship between velocity and acceleration, and basic gas laws.

Programs in the second class are more directly usable for education and training. These programs attempt to construct a model of the user by observing his or her interaction with the program. Some of these programs also embody teaching strategies, basing their suggestions and questions to the user on assumptions about how learning is best achieved. Systems that attempt to "understand" the user include: SOPHIE (Brown & Burton, 1975; Brown, Burton & deKleer, 1982), an "intelligent" CAI program that observes and evaluates a student's hypotheses as she or he tries to troubleshoot a faulted power supply; NLS-Scholar (Grignetti, Hausmann, & Gould, 1975), which attempts to judge and comment on users' interactions with a text editor (NLS) on the basis of their commands and make appropriate suggestions; and the WHY system (Stevens, Collins & Goldin, 1978), which teaches a student about the geographical aspects of rainfall distribution by initiating a Socratic dialogue, basing its questions on a dynamic, changing model of the student's comprehension.

Burton & Brown's (1979) version of the game, How the West Was Won, is a prototypical ICAI system. It incorporates both a model of the user and two distinct methods of teaching--coaching and modeling. Students play the game by forming an arithmetical expression out of three random numbers and using the resulting number to specify moves on a playing board. The "coach" built into the game observes the student's moves and makes comments such as: "If you had used parentheses on the move, you could have made the expression (2+3)*5, which would have moved you ahead to square 57, instead of your move, which took you to 49. Do you want to take the move over?" The computer also acts as a model by always choosing the best move on its turn.

Another such system is SOPHIE, an expert system that models an electronic power supply circuit. SOPHIE can be run in two modes: It can itself pose troubleshooting problems for a single person to solve, or it can be used as a game where one team sets a fault for another team to diagnose. In the first mode, the system sets a fault for the student to diagnose in a power supply circuit. The student can measure voltages and currents in different parts of the circuit (by asking the system questions) in order to figure out which component is faulty. The system evaluates the student's hypotheses about the fault by analyzing what it has told the student up to that point about the values in different components of the system and comparing these values to those that would occur under the student's hypotheses. This kind of comparison involves very sophisticated circuit simulation and fault propagation techniques. These same capabilities are used to tutor students in the team gaming option.

### 2.6.2 Glossary

CAI: Computer-Assisted Instruction; in general, refers to branching programs which present multiple-choice frames to students in an order determined by their previous answers.

CBI: Computer-Based Instruction; any teaching in which the computer plays a significant role.

CMI: Computer-Managed Instruction; primarily systems for keeping track of student's progress through a predefined set of examples.

ICAI: Intelligent Computer-Assisted Instruction; the use of AI techniques in educational software.

User Model: A dynamic picture of the student-user's knowledge in a particular domain, used by ICAI systems to decide on an educational approach.

### 2.6.3 State-of-the-Art

#### 2.6.3.1 Operational Applications

STEAMER (described above in section 2.6.1)

SOPHIE (described above in section 2.6.1)

BUGGY - simulates a student who has a consistent "bug" in his/her procedures for doing arithmetic. The user's task is to figure out what the procedural bug is.

DEBUGGY - generates possible underlying causes for manifestations of arithmetic problem errors.

All of these systems are operational in the sense that they could become commercial products without significant conceptual changes.

There are, of course, many demonstration systems; these tend to be incomplete in failing to cover the topic adequately, using a limited range of educational techniques, or being untested in real educational situations.

#### 2.6.3.2 Techniques that Make for Effective Operational Systems

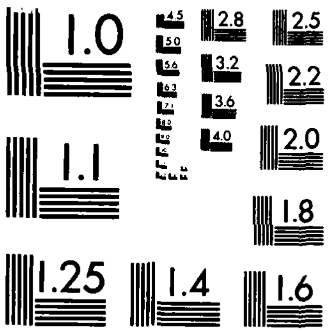
Teaching techniques which lead to effective operational systems include the following five. Each technique is exemplified by a program which uses it.

- Modeling: both static models (e.g., a solved example) and dynamic models (e.g., showing the problem-solving process).

  Example: SUMMIT is a program which displays numbers in a representation which makes clear their place value semantics. It then talks its way (by synthesized speech) through the process of adding or subtracting two numbers, making explicit the processes of borrowing, carrying, etc.

AD-A161 139 AUTOMATED INFORMATION MANAGEMENT TECHNOLOGY (AIM-TCCH): 2/3
CONSIDERATIONS FO (U) MACAULAY-BROWN INC FAIRBORN OH
W REITMAN ET AL MAY 85 AFAMRL-TR-85-042

UNCLASSIFIED F33615-82-C-0513 F/G 9/2 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

- o Coaching: Watch the student participate in the task and give her feedback and hints on how to refine performance.

  Example: How the West Was Won (described above in section 2.6.1)

- o Problem solving microworlds: Construct a sequence of problems which take the student from novice skills to expert skills.

  Example: Dynaturtle is a computer-based environment which provides a simulation of Newtonian dynamics. Students are given goals to pursue in games that help them correct their theories of force and acceleration.

- o Inquiry teaching: Forcing students to formulate theories by systematic questioning and then forcing them to debug their theories to account for more and more difficult data.

  Example: WHY system (described above) - helps students make explicit their models why and where rain falls, then presents a situation which challenges their model. The system is based on a semantic network-based model of facts about rainfall.

- o Explanation: Animation and verbal explanation designed to describe how a system operates. Negates incorrect models and can help a student transfer a correct model from a related domain.

  Example: STEAMER (described above in section 2.6.1).

All of these have been successful in demonstration systems. The best systems might use all five.

#### 2.6.3.3 Principal Areas of Research

Major emphasis recently has just begun on designing <u>programming environments</u> which are accessible to curriculum designers e.g.,(Programming By Rehearsal by Gould and Finzer at Xerox). In this case an environment has been built based on ideas from the SMALLTALK programming paradigm (see the chapter on AI Tools and environments), but it has been insufficiently used by non-programmer curriculum designers.

Continued research is highly likely to succeed (at least .8 probability) since results in very high-level programming environments already exist but have not as yet been applied to making computers more accessible to non-programmers. In our views, the designer will still be "programming," but at a level of detail much more appropriate to their basic expertise than currently available.

Fundamental breakthroughs making possible the design of systems without some "programming" are not feasible in the foreseeable future. It may be possible that some limited instructional domains will be simple enough that special purpose design environments can be created. One could envision a super version of SOPHIE, for instance, where one could input a particular circuit, constrained to be in a particular class. The instructor would not have to reprogram the system for each new circuit. The probability of such narrowly defined systems existing in the next 10 years is .6-.7, based on the effort involved in building such systems at present compared with the likely narrow scope of instruction.

In a real sense, an area of novel work is continually applying new results of AI and computer science in general to enhance the underlying system's capabilities. Speech synthesis (as in SUMMIT) and sophisticated graphics based on object-oriented programming (as in STEAMER) are two examples. Results in knowledge representation, planning, expert systems, problem solving, reasoning, language use, etc., will provide new potentials in computer coaching. This is ongoing work and will continue to be so. The likelihood of successfully applying new results in those areas is probably at least .7 given the success of current applications surveyed here. The limitations are those of the problems of research in knowledge representation, planning, expert systems, and natural language processing.

#### 2.6.3.4 Major Gaps and Problems

The biggest gap is between curriculum designers and technology; curriculum designers find current computer tools too inaccessible and sophisticated, while technologically competent people don't know enough about education to develop effective tutoring systems. See the previous section for a more complete discussion.

#### 2.6.3.5 Approaches that Failed

The "old" approach to involving computers in education, frame-based CAI, involved guiding students through a tree of multiple choice questions, with the choice of next question depending on the student's answers is unpromising except for low level educational tasks/goals. This approach has not taken advantage of the potential of the computer, nor has it proven educationally effective except in limited situations, such as the need to learn a large body of facts. Similarly, "intelligent" programs which simply adjust the level of difficulty of questions to an optimal level for the student (by choosing from a finite set) have not been notably successful.

#### 2.6.3.6 Major Laboratories and Key Contact Points

- BBN, Cambridge, MA/Allan Collins, Wally Feurzeig, Al Stevens, Barbara White
- XEROX PARC, Palo Alto, CA/John Seely Brown, Richard Burton, Kurt VanLehn
- Carnegie Mellon University, Pittsburgh, PA/Jill Larkin, Alan Lesgold
- Stanford University, Stanford, CA/Derek Sleeman
- Yale University, New Haven, CT/Elliot Soloway

#### 2.6.3.7 Recommended Key References


Brown, J.S., & Burton, R.R. Diagnostic models for procedural bugs in basic mathematical skills. _Cognitive Science_, 1978, _2_, 155-192.

Brown, J.S., & Burton, R.R. Multiple representations of knowledge for tutorial reasoning. In D. Bobrow & A. Collins (Eds.), _Representation and Understanding_. New York: Academic Press, 1975.

Brown, J.S., Burton, R.R., & de Kleer, J. Knowledge engineering and pedagogical techniques in SOPHIE I, II, and III. In D.H. Sleeman & J.S.

Brown (Eds.), *Intelligent Tutoring Systems*. London: Academic Press, 1982.

Brown, J.S., Collins, A., & Harris, G. Artificial intelligence and learning strategies. In H.F. O'Neil, Jr. (Ed.), *Learning Strategies*. New York: Academic Press, 1978.

Brown, J.S., & VanLehn, K. Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science*, 1980, *4*, 379-426.

Burton, R.R. DEBUGGY: Diagnosis of errors in basic mathematics skills. In D.H. Sleeman & J.S. Brown (Eds.), *Intelligent Tutoring Systems*. New York: Academic Press, 1982.

Burton, R.R., & Brown, J.S. A computer coach for "How the West Was Won." *International Journal of Man-Machine Studies*, 1979, *11*, 3-43. Also in D.H. Sleeman & J.S. Brown (Eds.), *Intelligent Tutoring Systems*. New York: Academic Press, 1981.

Clancey, W.J. Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies*, 1979, *11*, 25-49.

Collins, A., Warnock, E.H., Aiello, N., & Miller, M.L. Reasoning from incomplete knowledge. In D.G. Bobrow & A. Collins (Eds.), *Representation and Understanding*. New York: Academic Press, 1975.

Duda, R.O., Gaschnig, J.G., & Hart, P.E. Model design in the PROSPECTOR consultant system for mineral exploration. In D. Michie (Ed.), *Expert Systems in the Micro-electronic Age*. Edinburgh: Edinburgh University Press, 1979.

Dugdale, S. *Green globs: A microcomputer application for graphing of equations*. Urbana, IL: Computer-based Education Research Laboratory, University of Illinois, 1981.

Finzer, W. & Gould, L. Programming by rehearsal, *Byte*, 1984, *9*, 187-210.

Friedland, P.E. Knowledge-based experiment design in molecular genetics (Department of Computer Science Report No. 79-771). Stanford, CA: Stanford University, 1979.

Grignetti, M.C., Hausmann, C., & Gould, L. An intelligent on-line assistant and tutor -- NLS-SCHOLAR. Proceedings of the National Computer Conference, San Diego, CA, 1975, pp. 775-781.

Lindsay, R., Buchanan, B.G., Feigenbaum, E.A., & Lederberg, J. DENDRAL. New York: McGraw-Hill, 1980.

Pople, H. The formation of composite hypotheses in diagnostic problem solving - An exercise in synthethic reasoning. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977, pp. 1030-1037.

Shortliff, E.H. Computer-Based Medical Consultations: MYCIN. New York: American Elsevier, 1976.

Sleeman, D. & Brown, J.S., (Eds.). Intelligent Tutoring Systems. New York: Academic Press, 1982.

Soloway, E., Rubin, E., Woolf, B., Bonar, J., & Johnson, W.L. MENO-II: An AI-based programming tutor (Department of Computer Science Res. Report No. 258). New Haven: Yale University, December 1982.

Stevens, A.L. & Collins, A. The goal structure of a socratic tutor. Proceedings of the ACM Annual Conference, Seattle, WA, 1977.

Stevens, A.L., Collins, A., & Goldin, G. Diagnosing student's misconceptions in causal models. International Journal of Man-Machine Studies, 1978, 11, 145-156.

Stevens, A.L., Collins, A., & Goldin, G. Misconceptions in students' understanding. Journal of Man-Machine Studies, 1979, 11, 145-156. Also in D. Sleeman & J. S. Brown (Eds.). Intelligent Tutoring Systems. New York: Academic Press, 1982.

Stevens, A.L., Roberts, R.B., Stead, L.S., Forbus, K., Steinberg, C., & Smith, B.C. STEAMER: Advanced computer-aided instruction in propulsion engineering (BBN Report No. 4702). Cambridge, MA: Bolt Beranek and Newman, July 1981.

### 2.6.4 Summary

Preference in educational technique naturally results in varying goals in computer usage for educational purposes. A stereotype of traditional approaches to computers in education is automation of objective tests, e.g., multiple choice, fill-in-the-blank, and true-false questions, but with the powerful difference that the selection of questions can be made during the exercise based on the answers given to earlier questions. Obviously, systems can report percentages of answers correct/incorrect, which one were incorrect, etc.

The potential of AI techniques is in providing intelligent tutors which, as it were, observe the student's problem solving behavior, looking for flaws in their reasoning and methods. This offers the potential of a coach for learning problem solving, procedure, and reasoning. Furthermore, there is the potential of recognizing novel or rare mistakes in reasoning or in procedure instead of expecting all mistakes to fit in some predefined set of pigeonholes.

There have been several tours de force demonstrating this potential. Nevertheless, substantial problems to be solved are the great effort in creating an intelligent tutoring system and the gap between nonprogrammers and those who can create such systems. This area of AI is heavily dependent on continuing advances in natural language processing, expert systems, and knowledge representation.

## 2.7 PLANNING AND PROBLEM SOLVING UNDER REAL WORLD CONDITIONS

### 2.7.1 Overview

Classically, the goal of planning is to find a sequence of operations guaranteed to take you from some initial state to some desired end state. In the classical conception, effective planning was primarily a matter of search; research had to do with investigating the various search strategies (top down vs. bottom up, breadth first vs. depth first, etc). Problems having to do with the "planning policies" underlying search were generally left implicit.

"Planning policies" refers to those meta constraints determining acceptable planning procedures and solutions. The considerations involved here involve:

- o whether the planning must produce a solution that works under all conditions
- o whether it is merely reasonably likely to be workable
- o whether it works only under some explicitly specified assumptions, and conditions

In other words, "planning policies" refers to the need to make explicit our guidelines for determining the tradeoffs involved in processing costs vs. quality, unrestrictiveness, and "optimality" of the solution. Note the distinction between an optimal solution (cost what it may) and an optimal search procedure (which may be very cost effective, though it does not necessarily come up with the "optimal" solution).

Another type of issue under the general heading of planning policies has to do with what determines acceptable side effects. Some initial constraint, e.g., "object x is not movable," may in fact be one that is violatable, but only at some very large cost in the effort, or in side effects produced. Similarly, we need to make explicit the time restrictions we are operating under. A solution may be useless if it is only discovered well past the time it was required.

Still other similar planning policy issues involve specifying the resources we are willing to allocate, the restrictions on them, etc. Likewise, we need to make explicit what is an acceptable solution. For example, in planning in a game context, only the next move has to be specified unconditionally. Subsequent moves need not be.

In sum, once such planning policy questions are explicit, it is apparent that classical research on planning has been carried out under highly unrealistic assumptions with respect to real world planning conditions.

To see why, consider planning systems that would assist a military commander in command and control decision making. First, because the commander plans in order to achieve his own goals, he will benefit from a system that will assist him with the development of his own plans. Second, a system that achieves goals that a commander specifies must be a sophisticated planner in order to do what it is asked. To respond effectively, a system must plan in a complex domain involving both time and external events. Current planners cannot solve the planning problems that arise for such domains.

Why are current planners inadequate to the task at hand? Simply said, the knowledge representation systems that underlie current planners cannot be used to express many of the problems that are part of planning. As we have seen, classical planning research has for the most part focused on a restricted set of planning contexts. Goals and conditions typically are well specified. The objects and properties involved are known, and fixed. Nothing in the situation changes unless the user makes a change. There are no external events or agents, and no explicit representation of temporal relations. In contrast, in the domains we must consider, planning has to do with circumstances that are true at one time, or for some period of time, but not true at others. Hence, knowledge representation languages must provide representation for time and events. And planners must be designed to use these representations.

Not only are present languages lacking adequate representations of time and events, but also they cannot express goals and plans that vary in the degree of specification. Initially in planning, a user often has a vague objective that has few or no constraints. As the planning proceeds his objective becomes more constrained, sometimes to the point of being overconstrained. A representation capable of supporting planning must be able to add, delete, and transform constraints. That is, it must be able to deepen its descriptions progressively, and to transform the representation of desired actions and objects whenever it is unable to satisfy a current description. Current planners rely on representation systems that cannot express such concepts.

Since the user's objectives for his/her plans may initially be very underconstrained and may become over constrained, the planner must be able to act in the face of too little or too much information, to seek additional constraints or to relax some constraints. For example, in obtaining a display update, a user may request that a particular layout be displayed. Then he/she may request from the system.

"Show me how this relates to the plan I specified."

On the one hand, this request may be underconstrained because it is unclear just what relation the user has in mind or how the system is to illustrate the relation. A planner must have methods of recognizing the lack of information and determining how to proceed, either from its knowledge of the user and the context or from explicitly asking for clarification.

On the other hand, such a request may be made in a context that adds constraints to the request and in fact may overly constrain it. Thus, for example, constraints of scale in drawing a layout, or location of other display information may make it impossible to display the information desired. Sometimes the user may not have intended all these implied constraints. If the system is allowed to relax some of the implied constraints, the user may be quite happy with the result. Here, the problem is to devise a planning system that can use its knowledge to relax constraints appropriately, as an intelligent human assistant might be expected to.

In some cases, the set of constraints explicitly specified by the user may actually have no solution. If the user is to achieve any result at all, user and system must be able to explore ways of ascertaining priorities and evaluating subsets of constraints. This exploratory planning is a valuable tool for controlling the planning process. It allows the user to change a part of a plan, evaluate its results and then cancel the change and explore another part of the plan. Only in this way is the user likely to achieve an acceptable transformation of the original problem statement, i.e., one that redefines the problem but still satisfies his basic objectives.

Real world planning contexts may be subject to uncertainty, or to exogenously driven change. In situation assessment, for example, the information the planner works with may be inaccurate. Furthermore, the planning context is not under the planner's total control. He/she has to reckon with nature, and opponents. Thus, planning mechanisms are needed that can come up with useful results (1) in uncertain or changing contexts, and (2) in circumstances in which it is to be expected that the opponent will do everything possible to thwart these plans, and to advance his/her own.

Some of the main differences between classical and realistic research are listed below, by stating the new directions:

- representing time and events to support common sense notions and symbolic reasoning

- allowing for actions by other agents and for naturally-occurring events

- representing events whose occurrence overlaps in time

- modeling mental, as well as physical actions.

Another important aspect of planning that should be made explicit is the distinction between the underlying state space and the problem space. The state space consists of information about the state of the world and about the relationship of the possible operations to that information. The problem space, on the other hand, is associated with the particular planning methodology or planning discipline being used to find a state of the world that satisfies the planning objectives. Thus, for example, we can think of the planning space associated with a system like the General Problem Solver (GPS). This includes the particular basic operators that GPS makes available, the representation of the initial state and the goal specification, goal stack status, and the set of actions that can be applied (with information about their preconditions and outputs). Note that the operators of the planning space (or problem space) are not in general the same as the operators of the state space.

Once the initial effort to develop such disciplines as GPS had been accomplished, subsequent research attempted to explore the relations between descriptions in the state space and in the problem space. For example, depending upon the particular planning discipline being used, there might be a range of different descriptions in the problem space which correspond to a particular state in the state space, some descriptions being more useful than others.

Broadly speaking, the conceptual development of the field has proceeded from the first planning and problem solving system, GPS, to such subsequent generalizations as Sacerdoti's Noah system, in which the discipline of a strictly linearly ordered goal stack is replaced by the possibility of a partially hierarchical procedural net. This was followed by the MOLGEN system of Stefik, which tried to apply this generalized representation to metaplanning, making the decision about what subportions of the problem to work on next, a decision that could be planned about.

All of the systems just described work by breaking an overall problem into a conjunction of subgoals. This may be done recursively for each of the subgoals. One issue that becomes apparent when this view is taken is the question of how to handle the interactions among the conjoined subgoals. In particular, when you solve one subgoal, that may generate constraints that must not be violated in subsequent planning or problem solving work on other subgoals. The general approach that has been followed in dealing with this is to try to provide intelligent orderings of the subgoals.

Interactions can be thought of under two broad categories. There are interactions involving conflicts among the conditions assumed by individual subproblems. These have been studied for some time. However, there also are interactions having to do with the possibility that two subproblems, each of which can be solved without violating any of the planning policy constraints, will, when conjoined, violate such constraints (e.g., constraints on effort, constraints on time, etc.).

One pragmatic approach with dealing with these interaction problems, which has had some limited success in task specific planning and problem solving domains, is to define overall goal priorities and action preferences. In the long run, however, for intelligent planning and problem solving, it presumably will be necessary to endow systems with more flexible capabilities for discovering and dealing with harmful or interfering interactions among the subgoals.

### 2.7.2 Glossary

arc: (See definition of a graph.)

breadth-first search: exploring a state space by considering first solutions involving a single action, then those involving only two actions, etc. This is a technique which examines all alternatives before attempting to extend any line of action.

depth-first search: exploring a state space by considering first only one action, then a follow-on action to that, etc., considering an alternative first action occurs only if all extensions given the first have already been examined or eliminated.

goal: The statement of what is to be achieved. Viewing planning as a state space search, a goal identifies a number of nodes in the graph ("goal states")

graph: 2 sets, mathematically defined as a set of "nodes" (usually represented pictorially by circles) and a set of "arcs" (usually represented pictorially by arrows) which connect nodes.

node: (See the definition of a graph).

plan: a sequence of actions to achieve a goal. If one represents the alternatives as a state space, then a plan is a path. Sometimes a plan determines only the next step to take, sometimes it is a conditional plan with contingencies incorporated.

planning policies: conditions imposed on when a plan is acceptable and beyond merely achieving the goal. Examples include the cost and risk involved in carrying out a plan or the cost of searching for a plan.

problem space: denotes various states in the progress toward solving a problem. The transition from one state to another are problem solving maneuvers (of state space).

search space: another name for state space, based on the fact that finding a solution involves searching the graph for a path from initial state to a goal state.

solution path: a sequence of arcs in a state space leading from the initial state to some goal state.

state space: a graph where the nodes represent diverse states of the world and the arcs represent actions or operations that may be used to effect a change of state.

### 2.7.3 State-of-the-Art

#### 2.7.3.1 Operational Applications

If we define planning broadly enough, there are several systems that might be said to be operational planning systems. These include: the Digital Equipment Corporation XCON and XSEL systems, which configure VAXs and do planning at the time of taking a sales order. Additionally, there are systems which plan the synthesis of chemical compounds that are allegedly in operation at a number of chemical manufacturers, e.g., Dupont, Allied Chemical, Lederle Labs, and Hoffman-LaRoche.

#### 2.7.3.2 Techniques that Make for Effective Operational Systems

Current operational planners employ the same technique as in expert systems; see that section for those details. The reason is that expert systems also employ a search space in terms of several alternative rules (view them as "actions" changing the state of what is known) applying at a given time and in terms of many successive rule applications that may be needed to infer a conclusion.

#### 2.7.3.3 Principal Areas of Research

All of the principal areas of research share the framework described earlier of search through a state space, though the notion of "node" and "arc" may differ widely. The areas are stated below.

- Using various abstract search spaces above the level of concrete actions. The actions may be collections of concrete actions; the nodes may be generalizations of concrete states. The hypothesis is that examining the more abstract space (initially ignoring many details) will lead to general plans which may be refined into solutions, (See Sacerdoti, 1974; Vere, 1983)

- Studying alternatives to breadth-first and depth-first search. Many believe that measures of how near one is to finding a solution can be found so that numerical comparison enables the search algorithm to opportunistically explore alternatives. (See Hayes-Roth and Hayes-Roth, 1979; Pearl, 1983)

- Using distributed and parallel planning components. The techniques of dividing the planning process into components that can be executed in parallel and distributed over several machines is one way of capitalizing on the availability of microprocessors. (See Konolige & Nilsson, 1980)

- Providing for plan repair and incremental planning. A defective plan that is almost correct may need only a minor repair, namely, having special-purpose heuristics to identify the parts where repair is needed, and others to propose what repair to make. (See Wilkins & Robinson, 1981)

- Using explicit resource declarations with actions to account for constraints on the cost of executing a plan, the cost of finding a plan, etc. Optimization of a resource, if that is an issue as opposed to keeping resources below some threshold, is rather like using measures of nearness to the goal to guide search. Overlap in heuristic techniques for using such measures in this and the area mentioned earlier should not be surprising therefore. (See Pearl, 1983; Wilkins & Robinson, 1981)

Since all of these are highly exploratory, relatively new research endeavors, it is uncertain what the potential for success is. Given the newness of these endeavors that are based on an <u>established paradigm</u>, we estimate the probability of useful application at well above .5.

### 2.7.3.4 Major Gaps and Problems

Since planning has generally been simplified by considering only a single agent and an unchanging situation, an obvious gap is planning in dynamic environments, in which it is possible that the environmental situation may change, either due to natural causes, or to the activities of other agents operating in the situation. These issues are particularly important for planning in the military domain, where the environment may be under partial control of potentially hostile forces.

In these situations, the "planning policies" change. That is, one cannot guarantee that a plan that appears satisfactory at one time will be satisfactory at some other time (because the environmental constraints may change). However, one may nonetheless use planning to:

- o determine the significant, relatively invariant features of the environment
- o understand what their implications are
- o provide early alerts to significant changes in the environment that might affect current plans

Another gap involves developing plans in situations involving communication among multiple actors. Again, such planning research would be of tremendous and practical importance in a military context.

Still another gap is in planners that use both special purpose and general purpose methods appropriately. There often are well defined subproblems for which quite efficient algorithmic procedures can be used, for example, determining most effective routes in space. Universally applicable search strategies are general, but quite ponderous by comparison to a special purpose strategy. A really powerful planner would be able to recognize when it had a subproblem that could be solved using the more efficient special purpose strong methods appropriate to that subproblem, and rely on general-purpose reasoning strategies otherwise.

Additionally, research must be continued in the current areas listed earlier and in the following:

- o methods of providing more effective ways of coming up with appropriate problem formulations

- o better techniques for controlling search in realistic problem solving. Some candidates are listed below:

  - . decoupling strategic and tactical analysis
  - . focusing on specific goals and questions
  - . knowledge-based selection of options
  - . dynamic redefinition of relevant facts
  - . use of surrogates
  - . use of failure information to redefine goals

#### 2.7.3.5 Approaches that failed

There are several lines of work that have led to new knowledge about planning and new ways of thinking about planning, but which were not able to achieve the lofty goals for them. Planning directly in the state space is one example. Much of the early work on chess programs falls under this heading. Similarly, GPS might be said to have "failed" in the sense that it did not produce a "general" problem solver. However, it must be understood that most of this work was exploratory, and so to characterize it as a "failure" is to be overly literal.

#### 2.7.3.6 Major Laboratories and Key Contact Points

- o BBN, Cambridge, MA/N.S. Sridharan
- o SRI International, Menlo Park, CA/Stan Rosenshein
- o JPL (Jet Propulsion Laboratory), Pasadena, CA/Leonard Friedman
- o Carnegie-Mellon University, Pittsburgh, PA/Mark Fox, J.B. Carbonell
- o Rutgers University, New Brunswick, NJ/Charles Schmidt

o Simon Fraser University, Burnaby, BC, Canada/Nick Cercone

2.7.3.7 Recommended Key Refrences


Carbonell J.G. The counter planning process: A model of decision-making in adverse situations. (Department of Computer Science Technical Report). Pittsburgh, PA: Carnegie-Mellon University, February, 1979.

Fox, M.S. Reasoning with incomplete knowledge in a resource-limited environment integrating reasoning and knowledge acquisition. Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, B.C., 1981. pp. 313-318.

Hayes-Roth, B., Hayes-Roth, F. A cognitive model of planning. Cognitive Science, 1979, 3, 275-310.

Hayes-Roth, B., Hayes-Roth, F., Shapiro, N. & Wescourt, K. Planner's workbench: A computer aid to replanning (Technical Report). Santa Monica, CA: The Rand Corporation, 1981.

Konolige, K. & Nilsson, N.J. Multiple-agent planning systems. Proceedings of the First Annual National Conference on Artificial Intelligence, Stanford, CA, 1980, pp. 138-142.

Lehner, P. Planning in Adversity: A computational Model of Strategic Planning in the Game of Go. PhD thesis, University of Michigan, 1981. unpublished.

Pearl, J. Search and Heuristics. New York: Elsevier, 1983.

Reitman, W. Applying artificial intelligence to decision support. In M.J. Ginzberg, W. Reitman, & E.A.Stohr (Eds.), Decision Support Systems. Amsterdam: North Holland, 1982.

Rich, Charles. A formal representation for plans in the programmer's apprentice. Proceedings of the 7th Int'l. Joint Conference on Artificial Intelligence, Vancouver, B.C., 1981, pp. 1044-1052.

Sacerdoti, E.D. Planning in a hierarchy of abstraction spaces. Artificial

Intelligence, 1974, 5, 115-135.

Sacerdoti, E.D. A Structure for Plans and Behavior. New York: Elsevier North-Holland, 1977.

Sridharan, N.S. & Bresina, J.L. Plan formation in large, realistic domains. Proceedings of the Fourth National Conference of CSCSI/SCEIO, Calgary, Alberta, 1982, pp. 12-18.

Stefik, M. Planning with Constraints. (Report STAN-CS-80-784). Stanford, CA: Stanford University, January, 1980.

Vere, S.A. Planning in time: Windows and durations for activities and goals. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983, PAMI-5, 246-267.

Wilkins, D.E. & Robinson, A.E. An interactive planning system (Technical Note No. 245). Menlo Park, CA: SRI International, July 1981.

### 2.7.4 Summary

In AI, "planning" refers to the process of finding a set of actions which will transform some initial state of affairs to some desired state of affairs. As such, it will play an important part in expert systems of the future and also in natural language generation (to achieve goals in communicating). In general, planning research thus far has substantially simplified the problem by assuming there is only one agent that can affect the state of affairs and states do not change without an agent's acting. Of course, these are severe restrictions. "Planning policies" are additional constraints on acceptable plans to achieve a goal; these include time of completion, cost of carrying out a plan, cost of planning itself, safety of the agent, etc.

The major gaps in planning as a capability are:

- satisfying two or more goals. The problem here is that one cannot simply deal with the goals independently, since the plan to achieve one may undo the results of a plan to achieve another

- o taking planning policies into account
- o planning when there are multiple agents, some of whom can be adversaries
- o planning where the environment changes
- o determining how much effort to expend in trying to find a plan

## 2.8 AI TOOLS AND ENVIRONMENTS

### 2.8.1 Overview

The category "AI tools and environments" refers to hardware/software systems within which other AI research and development is conducted. It provides both the foundation on which AI work is built (i.e., AI programming languages) and the engineering environment in which that work is designed, implemented, and tested (i.e., AI programming systems). In this sense, it contributes to all of the other categories, from vision to natural language and speech, from knowledge representation to planning, and from training systems to expert systems. Yet it is properly a category in its own right with its own set of goals.

AI programs, almost by definition, are large and complex programs intended to perform complicated behaviors for which straightforward algorithms either are unknown (e.g., the comprehension of natural language) or cannot be computed with a reasonable amount of resources in a reasonable amount of time (e.g., playing chess). Consequently, the "solutions" to such problems are programs which at best approximate the desired behavior. Program development is very difficult and highly exploratory in nature. Historically, this has led to two orthogonal directions of research.

The first direction is in programming paradigms. Whereas early use of computers for scientific calculation motivated algebraic languages such as FORTRAN and ALGOL, the requirements of AI problems demanded languages with an essential symbolic character. The most important of these has been LISP, which embodies the functional paradigm. Among the other durable paradigms, one counts logic programming, object-oriented programming, and rule-based programming. More minor paradigms of past, present, and future include pattern match, constraint, and access centered schemes. Though many of these paradigms originated as special purpose notations arising from particular problems, many are being examined as general applicable languages because of their proven utility in special problems.

The second dimension is not programming language research itself, but the programming environment that supports the programmer in a given language. An interactive programming environment is built to support the language, including tools for four purposes: browsing, editing, debugging, and analysis. Briefly, browsing involves the presentation of information within the system (e.g., data structures, program components, analysis results); editing concerns the modification of the underlying representation of information through interaction with any of its many presentation forms (e.g., textual, graphical, structural); debugging controls the execution so that the details of program behavior can be observed and modified in order to achieve a correctly functioning program; and analysis makes explicit information (such as number of uses of a particular subprogram, computer time, etc. to solve a problem) which is otherwise only implicit in the static and dynamic relationships of program components and state. It should be clear that the tools in these separate groups are intimately related. (Many of the tools which were developed for AI have now been successfully applied to other programming languages; for example, many implementations of PASCAL now admit a degree of PASCAL-level debugging.)

If a programming system proves sufficiently successful, hardware can be designed and refined to substantially increase computational speed. This is important since AI applications tend to make intense demands on both computer time and memory. Such machines have been built to run LISP.

An important trend in the "AI tools and environments" category is the attempt to unify or integrate several of the paradigms within one system. The argument is that no single paradigm suffices for a sufficiently broad range of problems. Moreover, it is recognized that often current problems are merely components of larger issues and the component solutions will have to be integrated eventually. The goal is to find a conceptually clear way of joining paradigms together in order to provide a greater range of capability. In addition, this requires a proper abstraction of environment tools which can provide a uniform interface perspective over a larger scope of objects. An alternative, which has yet to be achieved, would be to find a truly unifying paradigm which singly captures the essential benefits of a number of the other paradigms. Whether this is even possible remains an open question.

Nearly all of this work has proceeded in the context of serial computation. The notion of parallel computation opens up new frontiers, but little has been achieved to date. A fundamental dimension of parallel computation is the size of the components comprising the parallel system, and their organization (network connectivity) is an open issue. Many hardware architectures have been devised along the size scale. AI problems typically require the subclass of such architectures which allow independent though communicating processes at each component. No applications have yet been achieved in parallel architectures for AI. There is large but untapped potential here.

### 2.8.2 Glossary

Access-oriented programming: programming where variables can be made "active", in the sense that read and/or write access to a given variable causes another program to run.

Browsing/inspecting: skimming (browsing) complex structure to focus (inspect) on a particular part of that structure.

Constraint programming: a form of AI programming based on performing a search, where constraints operate directly to block consideration of alternatives in violation of those constraints.

Debugging: the process of locating and correcting errors (bugs) in programs.

Functional programming: programming in a style that does not involve side-effects, such as changing the value of a variable. Its advantages are that it is far more amenable to verification, transformation, and parallelism. It is much closer in semantics to mathematical notion than to the semantics of side-effect programming languages, such as FORTRAN, PASCAL or ADA. Computational efficiency and ease of expression in purely functional languages are topics of debate at present.

Logic programming: programming using logical axioms as the instructions of programs. PROLOG is an example of a logic programming language.

Object-oriented programming: programming where procedures are organized around entities (objects) or classes of them. SMALLTALK is an example of an object-oriented programming language.

Pattern match programming: programming where subprograms are called not by giving their name, but by giving a pattern describing a goal to be achieved. The programs state what goal they apply to.

Pointing devices: input devices for identifying a particular spot on a CRT screen. Many computer workstations come with a "mouse," which is an example.

Rule-oriented programming: programming based on writing simple rules, such as if A & B & C then D.

Window systems: an input/output system where the display is divided into various rectangular regions (windows) so that i/o from various interrelated or disjoint activities may be visible at the same time.

### 2.8.3 State-of-the-Art

#### 2.8.3.1 Operational applications

The best operational examples of this work are the Interlisp and Zetalisp LISP systems, both of which are commercially available and which together support more AI research and development than any other programming environment. They provide not only robust implementations of their languages but an enormous set of programming tools. Common Lisp is an attempt to integrate the many dialects of MACLISP Zetalisp. Common LISP is available, though programming tools to support common LISP are still under development. Both Zetalisp and Interlisp are intended to support Common Lisp at some time in the future.

PROLOG is the most widespread language based on logic programming; several dialects and implementations exist and are widely used. Concepts from PROLOG are part of the basis of the Japanese Fifth Generation Computer Project. Use of Prolog in operational application is likely to grow.

LOOPS is a recent product from Xerox, and integrates functional, object, access, and rule oriented programming into one system. It is built on top of Interlisp; it is designed to support building expert systems. We expect its use in operational applications to grow.

SMALLTALK is the primary example of the object-oriented paradigm and provides a rather complete programming environment. The FLAVORS component of Zetalisp also embodies the object paradigm and is commercially available as part of that system. This paradigm is rather new. There is particular interest in it for applications in graphics, simulation, and CAI; see the chapter on tutoring and training.

#### 2.8.3.2 Techniques That Make for Effective Operational Systems

One factor for effectiveness is performance. If programs cannot be developed and executed in reasonable time, almost nothing else matters. The programming environments of the kind being discussed together with AI applications make intense demands. It is now typically cost-effective to dedicate a machine to a single user.

Robust environments with adequate tools are critical, since the program development task for AI programs is so demanding. This way investment can be shifted away from the implementation problem and more directly aimed at design issues and rapid prototyping.

Since program development in AI is demanding in that it requires breaking new ground constantly, the convenience of expressing things in the language and the degree of aid provided by the programming environment are critical to reduce the already large burden on AI programmers.

#### 2.8.3.3 Principal Areas of Research

As described in the overview, one principal area of research is in developing programming paradigms such as functional programming, logic programming, and object-oriented programming, including development of programming environments. Furthermore, integrating various programming paradigms into a single system (such as LOOPS) is an area of research.

Since earlier work has already led to operational systems such as INTERLISP and ZetaLISP, and since the concepts developed can often be incorporated into existing languages, the probability of the results of this research being applicable in AI programming is very high, say .9. The only limitation is as follows: though tools can lighten the burden of constructing AI systems, the software effort in constructing AI systems will be a burden for the foreseeable future.

There are additional research areas underway, namely workstations and parallelism. Discussion of them appears in the section 4.8.3 on milestones.

#### 2.8.3.4 Major Gaps and Problems

An ongoing problem is improvements in cost and performance of systems for AI programming. Advances in VLSI will continue to offer substantial improvements in both performance and cost.

The longer, harder problem is a conceptual one. The current programming paradigms are still not at a sufficiently abstract conceptual level; too much detail needs to be specified by the programmer. Consequently, the cycle time for trying new ideas is longer and more arduous than it might be.

Another long-term problem is the exploitation of parallelism. The interactions of many simultaneous computations are difficult or impossible for people to understand. One needs to find ways of aggregating parallel components such that the interactions between aggregates are minimized reducing conceptual complexity.

Only the very beginnings of effort to integrate several of the durable programming paradigms have appeared. It appears that no work is underway to create single paradigms which unify the essential characteristics of several of the paradigms. The distinction between integration and unification is an important one. Integration of several paradigms provides all of the selected paradigms with one setting together with mechanisms for aggregating them. Unification attempts to make available one paradigm whose components can provide at one time the capabilities normally found distributed among the several paradigms.

#### 2.8.3.5 Major Laboratories and Key Contact Points

- (Interlisp) Xerox PARC, Palo Alto, CA/Beau Sheil, Larry Masinter

- (Zetalisp) Symbolics, Cambridge, MA/Daniel Weinreb
- (SMALLTALK) Xerox PARC, Palo Alto, CA/Adelle Goldberg, David Robson
- (PROLOG) Quintus Computer Systems, Palo Alto, CA/David Warren
- (Multi-LISP) MIT, Cambridge, MA/Burt Halstead
- (Programmer's Assistant, parallelism) MIT, Cambridge, MA/Charles Rich, Carl Hewitt
- (Common Lisp) Tartan Labs, Pittsburgh, PA/Guy Steele
- (Common Lisp) CMU, Pittsburgh, PA/Scott Fahlman
- (IBM Lisp) IBM, Thomas J. Watson Research Center, Yorktown Heights, NY/Richard Jenks
- (LOOPS) Xerox PARC, Palo Alto, CA/Daniel Bobrow, Mark Stefik, Sanjay Mittal
- (parallelism) Bolt Beranek and Newman Inc., Cambridge, MA/Don Allen, R. Rettberg, N. Sridharan

#### 2.8.3.6 Recommended Key References


Clocksin, W.F. & Mellish, C.S. Programming in Prolog. Berlin: Springer-Verlag, 1981.

Goldberg, A., & Robson, D. Smalltalk-80: The Language and its Implementation. Reading, MA: Addison-Wesley, 1983.

Sanella, M. (Ed.). Interlisp Reference Manual. Palo Alto, CA: Xerox Corp., 1983.

Sheil, B. Power tools for programmers. Datamation, 1983, 29, 131-144.

Stefik, M., Bobrow, D., Mittal, S., & Conway, L. Knowledge programming in LOOPS. AI Magazine, 1983, 4, 3-14.

Weinreb, D. & Moon D. Lisp Machine Manual. Cambridge, MA: Symbolics, Inc., 1981.

### 2.8.4 Summary

The term "AI tools and environments" refers to the hardware and software provided for research and development of AI. That programming environment is particularly critical to AI since:

- AI systems tend to be very large (in terms of number of lines of code)
- AI research and development centers on devising systems that have not been built before
- as a consequence of the two above, AI is very labor intensive
- AI applications and prototypes typically make intense demands for computer time and main memory
- AI research often involves much empirical use of prototypes to evaluate their effectiveness

Typical of the environment of choice for AI research at present is a powerful "workstation." This involves a computer designed to serve a single user at any time, so that the intense demands for computer time and main memory are met. They normally involve:

- a fast processor comparable to a mini or super mini-computer
- 1-8 megabytes of main memory (This is also comparable to a mini-computer.)
- local disk space of 20-500 megabytes
- a graphics console
- sophisticated display management
- a rich dialect of LISP as the programming language and interface
- network capabilities to provide for large file storage and printing

Research in this area falls into three categories:

- generally commercial research in order to provide more powerful, less expensive workstations

- hardware research in highly parallel computers

- programming environments, particularly to support parallel computations (In general, parallel programming is a little understood, very difficult research problem.)

## 2.9 SPEECH

### 2.9.1 Overview

Speech recognition may be defined as deriving the linguistic message from a spoken utterance. The term is also used in contradistinction to _speech understanding_, where _speech recognition_ refers to deriving only the words that were spoken (such as for a "phonetic typewriter), and _speech understanding_ implies building a representation of the meaning of the utterance as part of the recognition process, which representation is then used as part of a person-machine interaction task (Newell et al., 1973; Walker, 1973; Wolf, 1980). In this report, this distinction is not especially important, and we shall use the term _speech recognition_ in its general sense.

This definition of speech recognition depicts it as the mechanical equivalent to the human ability of speech perception, and therefore it is necessary to focus on the important dimensions along which speech recognition systems lie. These dimensions are:

- isolated words vs continuous speech

- speaker dependence

- vocabulary or language complexity

- conditions on the acoustic environment and on the speaker

- speed of operation

We treat these subjects in more detail below.

### 2.9.2 Glossary

Isolated/Continuous: Isolated word recognition (IWR) refers to the recognition of words or phrases spoken in isolation, i.e., delimited by silence. Words thus spoken are not affected by the context of neighboring words ("did you" vs. "didjew"), and the silences make the word boundaries easy to spot, so the recognition is made much easier. Connected speech recognition, on the other hand, is much harder (and requires more computation) because of phonological and phonetic word boundary effects, and because the boundaries between words are not clearly marked in the acoustic signal, they must be inferred. The earliest commercial speech recognizers were isolated word recognition. Even today, only a few CRS systems are available, and they are much more expensive than isolated word recognition systems.

Speaker Dependence: Each person produces a different speech signal, due to differences in anatomy, dialect, and idiosyncrasies. This diversity is handled with apparent ease over wide variations by humans, but neither this ability nor the personal differences in the signal are sufficiently well understood. Performing speech recognition in a speaker-normalized or speaker-invariant manner has proved to be a challenge, even over a narrower range of variations. Simple speech recognizers are speaker-dependent, in that they must be "trained" with speech samples of each vocabulary item by the speaker; a different speaker requires his own training patterns. Several approaches to partial or full speaker-independence have been investigated, but even the most successful ones operate over only a limited domain. Speaker independence remains an important but elusive goal. (The term "speaker independent" deserves, but rarely receives, qualifications. As a practical matter, it cannot include literally every speaker of the language. Relevant questions are: Does it include both men and women? Children? One dialect only or wide variety? American English speakers only or foreign accents also? Even among speakers of the same dialect, there are a few that seem not to perform well with speech recognizers [Lea, 1980, p. 561]).

Complexity: The complexity of a speech recognition task is not easy to define or measure. For small vocabularies, it depends on the size and makeup of the vocabulary (a larger vocabulary, shorter words, and words that are phonetically similar are more difficult to recognize). However, in a large vocabulary, where vocabulary makeup is not controllable, system performance is largely related to the complexity of the allowable linguistic structures. (Here we introduce the notion that real applications employing large vocabularies must have grammatical constraints. Allowing any word to appear anywhere in an utterance is not communication but would be nonsense, and any recognizer that fails to use such constraints is working on an artificially difficult problem!) Vocabulary size is not directly important, for the grammatical complexity determines the number of possible words at each point in the grammar.

Environmental and Speaker Effects: The quality of the speech signal, as determined by the absence of noise, interfering signals, and distortion, is important for speech recognition. If a task must be performed in a high noise environment (such as in a vehicle or factory) or under variable transmission conditions (such as over the telephone), these effects on the signal will make it more difficult to recognize.

Utterances produced by speakers subject to variable health (e.g., nasal congestion), emotional stress (e.g., excitement or danger) or physical stress (e.g., exertion or g-force) contain significant additional variability that must be handled by speech recognizers, adding to the difficulty of the recognition task.

Prosody. Prosody is acoustic information above the level of segments, for example, stress, timing, inflection, and pitch.

Speed of Operation: Human perception of speech is virtually instantaneous once the speech has been uttered. This rapidity of communication is one of the attractive aspects of speech for person-machine interaction, but it places a severe constraint on speech recognition systems, to operate with roughly the same speed as the speech is produced. Many research systems, of course, do not achieve this speed, but they must do so eventually if they are to become practical. Advanced computation, e.g. fast processors and parallel processing, must be available at low enough cost for complex speech recognition ever to be practical.

### 2.9.3 State-of-the-Art

The first commercial speech recognizers (limited vocabulary, speaker dependent, isolated word recognition systems) appeared over 10 years ago, and the number of commercial products has burgeoned as recognition techniques have been refined and as computational ability/cost has increased. This commercial presence provides a convenient criterion for distinguishing operational applications from demonstration or research systems. (The commercial boom has also been matched by the number of industrial concerns performing research; unfortunately their results and techniques are often not always available.)

A prime difficulty in comparing systems is that system performance depends on task difficulty, which as stated before, is not directly measurable. Even when a vendor or researcher quotes performance results, they refer to a specific set of conditions, and it is frequently unclear how the system would perform on a second set of conditions: different vocabulary, speakers, noise conditions, etc. Standardized performance testing is a current area of research and development.

Commercial speaker-dependent isolated word recognition systems offer vocabulary sizes of 20-150 words (and higher) at costs of $1-10K. An exception to this is software available from Dragon Systems, Inc. at a $10 per unit licensing fee and which operates on an 8088 or 6502 based personal computer. Some systems claim to handle noise or telephone input. Recent tests on a common 20-word vocabulary show error rates between 13% to 0.2% in quiet and 30% to 0.5% in moderate noise (Lea, 1980), so performance of some systems is poor. Such systems generally use a filter bank to do a short-time spectral analysis of the speech and model the words as patterns of energy in time and frequency. Recognition is performed by comparing such patterns without analysis of phonetic units. Some systems use dynamic programming to achieve a time alignment between input signal and stored patterns. A few systems claim speaker independence, but only on very small vocabularies.

A few commercial speaker-dependent CRS systems are available, in a restricted sense known as connected-word recognition. In connected word recognition, word models are "trained" in isolation (and in one system, they are refined by training from connected word utterances). Recognition uses the same sort of short-time spectral analysis and an elaboration of the dynamic programming time alignment used in isolated word recognition systems. At much greater computational cost, this process can deduce the word boundaries, but it can do little about word-boundary effects. Therefore the vocabulary items should be phonetically dissimilar, and the input speech should be somewhat carefully enunciated. At least one connected word recognition system allows grammatical constraints and several hundred word vocabularies.

In research laboratories, grammar-directed isolated word recognition and connected word recognition systems are more common. One approach to speaker independence uses multiple templates per word and training with exemplars from many speakers followed by clustering and merging of similar templates. This, or course, requires additional computation.

Connected word recognition systems seem adequate for many applications of low complexity, but they cannot be easily extended to very large vocabulary, high complexity tasks. The training of each vocabulary item from exemplars becomes impractical, linguistic knowledge (such as between-word contextual effects, fluent-speech phonological effects, and dialectal effects) cannot be handled adequately, and phonetic knowledge of speech cannot be applied at all. Consequently many laboratories are developing connected speech recognition systems based on smaller linguistic units such as phones, diphones, in-context, demisyllables, or syllables. The problem is still difficult, for while there may be fewer phones than words, the phones are severely affected by context (coarticulation). The methods used for modeling and recognizing these units range from traditional acoustic-phonetic features to syntactic pattern recognition to statistical models such as hidden-Markov and to combinations of these.

The preceding discussion has focused on speech recognition at the word level and at the subword level. The use of simple grammatical knowledge is becoming more common, but complex grammars, such as natural language subsets, still lie in the future. The use of other knowledge sources, such as phonological rules, prosodics, semantics, and pragmatics, which was espoused and initiated during the DARPA Speech Understanding Project of the 1970s (Newell et al., 1973; Walker, 1983), has largely lain dormant since then. Multiple knowledge sources cannot apply themselves; strategies for applying diverse multiple knowledge sources are themselves a research topic. Without changes in funding, this particular area of all the areas is likely to remain unresolved.

Significant computation will be required to achieve high performance in large vocabulary, high complexity applications, and there is potential parallelism in many speech recognition paradigms. Therefore speech recognition is a good candidate for implementation in a multiprocessor computation environment.

Another area of increasing difficulty is that of measuring system performance as system capabilities increase (e.g., large vocabulary, continuous speech, many speakers) and as system error rates become close to zero. This is a problem both for the researcher ("How can I tell if my last change was an improvement?") and for the marketer. The amount of speech required for training and testing is large, as is the number of system operations that must be observed. Automatic testing over extended periods of time is required.

#### 2.9.3.1 Major Laboratories and Key Contact Points

Many laboratories and companies are active in speech recognition research at several levels. Some of this research is oriented toward short-term products, and some has longer term goals. Some of the major laboratories in this country are:

- AT&T Bell Laboratories, Murray Hill, NJ/James Flanagan
- BBN Laboratories, Cambridge, MA/John Makhoul
- Carnegie-Mellon University, Pittsburgh, PA/D. Raj Reddy
- Dragon Systems, Newton, MA/James Baker
- Fairchild Lab for AI Research, Palo Alto, CA/Richard Lyon
- IBM T.J. Watson Research Center, Yorktown Heights, NY/Frederick Jelinek
- ITT Defense Communications Division, San Diego, CA/Robert Wohlford
- Massachusetts Institute of Technology, Cambridge, MA/Victor Zue, Jonathan Allen
- SRI International, Menlo Park, CA/Jarad Bernstein
- Texas Instruments, Dallas, TX/George Doddington
- Verbex Corporation, Bedford, MA/Chris Seelbach

There are other major laboratories abroad, notably in Japan and France.

#### 2.9.3.2 Recommended Key References

Bahl, L.R., Jelinek, F., & Mercer, R.L. A maximum likelihood approach to continuous speech recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983, PAMI-5, 179-190.

DeMori, R. & Laface, P. Use of fuzzy algorithms for phonetic and phonemic labeling of continuous speech. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1980, PAMI-2, 136-148.

Dixon, N.R., & T.B.Martin (Eds.). Automatic Speech and Speaker Recognition. New York: IEEE Press, 1979.

Haton, J.P. (Ed.). Automatic Speech Analysis and Recognition. Dordrecht, Holland: Reidel, 1982.

Lea, W.A. (Ed.) Trends in Speech Recognition. Englewood Cliffs, NJ: Prentice-Hall, 1980.

Lea, W.A. Selecting the best speech recognizer for the job. Speech Technology, 1983, 1, 10-29.

Newell,A., Barett, J., Forgie, J.W., Green, C., Klatt, D., Licklider, J.C.R., Munson, J., Reddy, D.R., & Woods, W.A. Speech Understanding Systems: Final Report of a Study Group. Amsterdam: North-Holland Publishing Co., 1973.

Simon, J.C. (Ed.). Spoken Language Generation and Understanding. Dordrecht, Holland: Reidel, 1980.

Walker, D.E. Speech understanding, computational linguistics, and artificial intelligence (Artificial Intelligence Center Note 85). Menlo Park, CA: Stanford Research Institute, 1973

Walker, D.E. (Ed.). Understanding Spoken Language. NewYork: Elsevier North-Holland, Inc., 1978.

Wolf, J.J. Speech recognition and understanding. In K.-S. Fu (Ed.), Digital Pattern Recognition. Berlin: Springer-Verlag, 1980.

### 2.9.4 Summary

Speech recognition and synthesis is certainly one of the most exciting potential applications of AI, both because of the added dimension in natural communication with computers and also because of its need in certain environments, such as a cockpit. For many applications, there already are adequate synthesis systems available. Speech understanding is lagging behind. Furthermore, the most difficult problems in synthesis remain in understanding as well, such as prosody. As a consequence, our report has focused only on speech recognition.

It is important to distinguish between isolated word recognition, where there is clear silence between words, and continuous speech recognition, where there is not. In continuous speech the adjacent words affect the sound of the current word, e.g., making "I scream" and "ice cream" impossible to distinguish phonetically. Virtually all commercially available systems are isolated word recognition. Other difficult problems include variation among speakers within the same dialect, variation across dialects, variability in an individual speaker (e.g., due to stress), level of background noise, vocabulary size, and grammar simplicity/complexity.

We are many years from being able to have truly natural speech input. In addition to the significant problems of deciphering speech, there are also the problems of natural language understanding as discussed earlier in section 2.3.

## Section 3. FUNCTIONAL SPECIFICATIONS FOR FUTURE SYSTEMS

Mark E. Jones
Kenneth R. Boff, PhD
Ralph Weischedel, PhD

### 3.1 INTRODUCTION

A three-day workshop was conducted during April-May 1985 in the three AFAMRL technical domains: systems design, piliot/aircrew automation; and command, control and communications ($C^3$). Each technical domain was treated independently by different working groups. The workshop used a variety of specialized techniques to stimulate the flow and exchange of ideas and information among participants with the objective of identifying control and information management choke points in each domain. The documented output of each working group was distilled into a listing of functional specifications which were candidates for application of aritificial intelligence based technology. By functional specifications, we mean descriptive statements of system performance necessary to overcome the identified choke points. These were grouped in each technical domain into three categories.

1. Communication. Approach to input/output between the user and machine subsystems.

2. Expert understanding. Embodiment of and adaptive accessibility to domain knowledge.

3. Decision Aiding. Use of domain knowledge to support control/information management operations of human operators.

Because of the apparent similarity of many of these functional specifications across the three technical domains, contextual scenarios were developed for each technical domain which would enable differentiation among these specifications. These scenarios were provided to BBN for analyses of the AI capabilities necessary to support the technology demands of the scenarios.

## 3.2 SYSTEMS DESIGN

### 3.2.1 Functional Specifications

A. Communications (Input/Output)

- Designer interacts with the system using a variety of inputs including natural speech, keyboard, etc. The system may respond with visual displays, natural language, etc.

- System queries the user at an appropriate level of detail adaptive to the user's level of understanding.

- System allows access to successive levels of detail.

- System allows feedback from data base users and from end users of product designs.

- System cues the user to the availability of additional information on a given topic.

- System has rapid turnaround.

B. Expert Understanding

- Expert "understanding" of domains (i.e., technology, methodology, design process, jargon, etc.) of designers, psychologists (i.e., research psychophysicists) and human factors practitioners.

- Interacts intelligently with other relevant data bases (official specifications and guidelines, materials, previous designs, lessons learned, data bases in psychology and medicine, etc.)

- Intelligently filters interactions with user in interface with data base information.

- Able to learn and expand data base.

- Intelligent problem reduction.

C. Decision Aiding

- Helps user define objective of search.

- Provides comprehensive tailored response.
- Capable of rapid generation of prototypes and demonstrations.
- Intelligent data compression and manipulation.
- Data interpolation and extrapolation.
- Infers data reliability with respect to design application.
- Provides auditable track of process.
- Rapid turn around.
- Provide directory of researchers and specialists who might offer consultation on the problem under consideration.
- Provide information needed to make trade-offs at progressive stages of development.

### 3.2.2 Systems Design Scenario

The expert design terminal (EDT) is a computer terminal with artificial intelligence (AI) interface capabilities, designed, and manufactured for the DoD systems design community. A network of EDT's has been installed with individual units at design related facilities, and any number of the EDT's may be data-linked. Voice communication among the operators (i.e., the design team) at the terminals is provided via a tele-headset. Each EDT is situated in a secure and environmentally controlled cubicle. The peripheral equipment includes: two displays (viz., a holographic viewer and a CRT screen) having multi-color presentation capabilities; a letter-quality/color graphics printer; a keyset consisting of a keyboard with a cursor trackball, a numeric keypad, and program function keys; a hybrid (i.e., digital and analog) graphic reader for printed material in microfiche or hardcopy; a natural language voice recognition and synthesis communication system; a high-resolution video camera capable of transmitting any scene to another terminal for display and/or printout.

A design engineer sits down at an EDT to perform a front-end analysis for design of a pilot trainer simulator for the latest AI-interfaced advanced fighter/attack aircraft (a/c), the F-34 Magicman. The F-34 has already been designed and is undergoing a prototype evaluation. The designer activates the EDT and then uses the keyset to enter his/her social security number. The EDT then scans its identification file for his/her educational background, specific areas of expertise, and past design experiences with the EDT. While scanning this information, the EDT asks the designer to describe what it is that he/she wishes to accomplish. "To define training simulator design requirements and specifications for the F-34 Magicman a/c," replies the designer. "What are the training objectives you are trying to meet?" asks the EDT. The EDT is informed that the trainer will initially be used to facilitate the transitioning of experienced pilots and the training of new pilots for the F-34. Later the trainer will be used to maintain proficiency of these F-34 pilots. The EDT asks for a brief example of the types of flight missions that will be trained. It is told that the training scenario should include take-off, fly evasive maneuvers at low altitudes, deliver weapons on a target, troubleshoot malfunctions, and return home safely.

The EDT, understanding design jargon, responds by listing system fidelity and transfer of training constraints on the design. The designer considers these and verbally raises concerns about how these will tradeoff against budgetary constraints. The EDT can understand and appropriately incorporate the designer's concern as it was expressed in "fuzzy" logic. The EDT subsequently provides verbal reassurance to the designer that financial constraints will be properly weighted. The EDT, using its stored knowledge of past design experiences with this designer, knows that he/she likes to be presented a listing of available data bases related to the design requirements. The EDT displays and prints out relevant design data bases on simulation (e.g., specifications, standards, guidelines, materials, and lessons learned) and human factors data bases (e.g., physiological, anthropological, human performance/perception, and psychological data).

In a subsequent phase of the design, the EDT is requested to display a holographic model of a pilot in the flight station of an F-34 operating the controls as would be done during an attack mission. As the EDT does this, by accessing a data base network on operational characteristics of jet fighters, it also prints out two-dimensional sectional views (viz., front, top, side, back, and/or auxiliary) for each pilot movement, which show the anthropological data involved for the fifth to ninety-fifth percentile pilot group. It also generates tentative Instructional Systems Development (ISD) data for these actions. These new data are inferred using the EDT's ability to intelligently interface relevant data bases (e.g., ISD guidelines and operational procedures) with the anthropological measurements which have been generated. While this activity is occurring, the EDT analyzes the displayed situational workload to discern choke points that should be focused on during pilot-training sessions. This information is tailored to the design unit measurements and terminology that the designer will understand.

Since there are already two operational prototypes of the Magicman, the designer realizes that engineering change proposals (ECP) will have to be accounted for in the simulator design. The EDT is requested to link-up with the Magicman System Program Office (SPO) data banks to search for ECP's. The EDT prints out seven ECP's, does a high-level analysis of these and displays the probability rating of "high-likely," "fairly-likely," or "not-at-all" of each ECP affecting the designed simulator's effectiveness. Each ECP listed is also accompanied by its status (i.e., accepted, rejected, being reviewed, or awaiting review). The designer is confused by one particular ECP and indicates the need for elaboration by moving the CRT cursor, with the mouse, to the ECP and depressing a function switch on the keyset. The EDT displays the questionable ECP's documented sections containing the "action" and "rationale" statements. The EDT also cues the designer that a point-of-contact (i.e., the originator) for this ECP is available, and how the originator can be reached if clarification at another level is necessary.

The designer tells the EDT to update the progress of the simulator design after reconfigurations have been made using the ECP's. The EDT responds that the design plans for the student/operator flight station are complete with regard to the location of the associated equipment used by the pilot. System realism for perceptual cues (i.e., visual, auditory, and motion information) in the student/operator station needs to be designed with respect to physical and/or perceptual fidelity requirements and constraints. The designer requests that fidelity requirements, showing the probability estimates for transfer of training of the simulator to the actual a/c, be set for 85% with regard to the constraints imposed by the EDT, unless otherwise instructed. The EDT manipulates data on control handling and determines, by intelligent problem reduction, the fidelity of components in the student/operator station of the trainer. Next, the system indicates that the instructor/operator station needs to be designed and functionally located with respect to the student/operator station. This location will allow the instructor to give guidance quickly, while monitoring each student's activities unobtrusively. Further analysis, using the holographic model, must be done on the pilot's workload, data interpretation, and performance. The EDT then prints out a directory of available (i.e., with an open contract) training psychologists and human factors specialists that could be consulted. The designer appreciates this, and informs the EDT that he/she does not know who is best to contact.

The EDT recommends four specialists. Two perceptual psychologists that could assist in design of the visual equipment needed for retinal display of the real world field of view and computer generated symbology, and two human factors experts to support the functional allocation (e.g., purpose and frequency of use) of control panel design and workplaces. The EDT suggests that they could be useful in optimizing the instructor/operator station panel and simulator layout. This information was deduced from the EDT's comparison of the F-34 flight systems simulator with other previous simulator designs. The EDT is directed by the designer to communicate with these specialists to find out if they are able to link-up with an EDT to become part of the design team.

The designer dons the EDT's tele-headset and talks with a training psychologist. The psychologist proposes to determine the needed perceptual and motion specifications, the simulator functional layout, and transmits them to the designer. The psychologist also suggests that a portable EDT-linked analog/digital recorder and monitor be integrated into one of the F-34 prototypes, which could collect data during tactical exercises. This data could then be used to provide the simulator's real-world computer generated signals, and to update the EDT's fidelity estimates pertaining to tactical mission problems. The designer agrees and ends his conversation with the psychologist. The designer then arranges to follow through on the psychologists suggestions, after requesting the EDT to begin transmitting the appropriate contract for these formal activities to the psychologist and to notify the designer's contracts office about these agreements.

The EDT is requested to prepare a scheduled time-frame in which the trainer would be fully operational (i.e., assuming no problems in production and set-up) at the various training facilities; and to indicate critical paths with respect to the constraints of time, money, and people. The EDT estimates the task completion date as three weeks behind the first F-34 delivered to any command. The designer is amazed, for since the development of the EDT, the lag time between the delivery of a new a/c and its associated trainer has been significantly reduced.

### 3.2.3 Analysis of Systems Design Scenario

In the systems design scenario, the following capabilities are required.

- Speech understanding and synthesis in a broad domain (para. 1 and throughout). The designer can use a free form in speech. For instance, in para. 2, the designer combines two goals in one utterance with no obvious cue connective to state their relationship, rather than having to restrict him/herself to selecting one pre-defined goal. The domain is not limited since the topics include design goals (para. 2); flight scenarios (para. 2); data base contents of diverse data bases (para. 3, 4, and 5), (e.g., SPO's, para. 5; simulator design and characteristics; specialists who can advise on design, para. 7; contracts, para. 8; and schedules for task completion, para. 9).

- Natural language understanding in a broad domain (throughout). The reasons are the same as those given above for speech understanding. Note that the designer can use typed input, as well as speech.

- o Natural language generation (para. 2). Note that the purpose here is to paraphrase the system's understanding of the designer's requests. This is demanding in that the system must express its understanding highly precisely; otherwise, any system misunderstanding of the designer will not only go unnoticed but also may be compounded.

- o Expert advice on relevant data bases (para. 3) and who can help (para. 7). Note that this assumes knowledge of the task, knowledge of data base contents and their purposes, a model of the designer's goals, and a model of the needs and capabilities of individuals.

- o Representation of space (para. 4). This is a model rich enough to support planning pilot movements and detailing cockpit layout not only for displaying simulations of pilot behaviors but also for determining possible choke points.

- o Representation of tactics of an aircraft (para. 2 and 4), including the details of pilot actions in those tactics.

- o Reasoning about bodily movements (para. 4), both as a part of pilot actions and as a source of problems in instruction (para. 4), as well as choke points (para. 4).

- o Reasoning about what changes in cockpit design affect simulator design (para. 5).

- o Reasoning about fidelity of simulator design (para. 6).

- o Analogical reasoning about designs and about plans (para. 7). This is critical to determining what previous simulator specifications may be of interest to the designer.

- o An expert system to write contracts (para. 8).

The most important facets to note about these requirements are:

- o the breadth of knowledge and reasoning capabilities

- o integration of knowledge from several data bases from potentially differing views, e.g., human factors, psychology, engineering specifications, light tactics, and contract constraints

- o the sophistication of natural language (including speech) understanding and generation

More than anything else, these three factors govern how far away such a designer's associate is.

## 3.3 PILOT/AIRCREW AUTOMATION

### 3.3.1 Functional Specifications

A. Communication

- The system uses natural language understanding and synthesis to communicate with the pilot.
- The physiological state of the pilot, the operational status of the aircraft systems, and the environmental situation surrounding the aircraft are sensed and communicated to the pilot.
- The system evaluates the pilot state/workload with respect to impedance matching of communication before and during aircraft to pilot interaction.
- The system cognitively matches the pilot for the best method of information portrayal.
- The system is able to communicate or interact with the pilot in a multi-modality manner (e.g., visually, aurally, and/or tactually).

B. Expert Understanding

- The system utilizes preprogrammed (i.e., current situation) information to understand:
  - mission and tactics
  - pilot (psychology, physiology, training, and experience)
  - the aircraft systems (status, control dynamics, offensive/defensive, kill mission completion)
- The system integrates new information (i.e., updates) into its programmed network.
- The system allocates information in a favorable manner to the pilot with respect to workload.

- The system learns and adapts to changing status of the pilot, aircraft systems, and environment.

- The pilots' workload is monitored respective to sensing, planning, decision making, and execution during the mission.

- The system can assess situations for pilot safety, mission completeness, and when acting as a $C^3$ platform.

C. Decision Aiding

- Adaptive to varing levels of control/automation in response to pilot instruction and/or workload.

- Adaptive executive/subordinate roles with regard to pilot instruction and/or mission goals.

- Maintains a constant prioritization of mission objectives and threat situations.

- Error analysis during operations for determining "risk levels" of desired maneuvers from a comparison between:

  - pilot to a/c

  - a/c to pilot

- Confidence levels for uncertainty for mission strategies.

### 3.3.2 Pilot/Aircrew Automation Scenario

A fighter pilot enroute to the assigned target area requests his/her aircraft (a/c) to perform a systems check and an environmental surveillance. The a/c is instructed to maintain this activity throughout the mission. The pilot requests a probability profile (i.e., a readout) of the appropriate tactics which might be used to successfully complete the mission.

The a/c's expert knowledge base system has been programmed with the complete training, medical, psychological profile, and educational history of the pilot. It also contains the in-flight records of all the pilot's prior missions with this a/c type. This special information, and the generic knowledge base of fighter pilot tactics, both offensive and defensive, added to the current information from its physiological, environmental, and system status sensors enable the a/c to respond quickly and accurately, in a specifically tailored manner, to the pilot (e.g., data presented in an order that the pilot favors). The a/c computes the current workload of the pilot, and together with the knowledge of the mission, tactics, and the current situation, prioritizes the information it is required to relay audibly and/or visually, and simultaneously adjusts the a/c's aerodynamic structure (e.g., the a/c might be reconfigured for increased maneuverability when nearing a hostile area).

The pilot then asks the a/c to report its current assessment of their situation (i.e., the mission tactics, goals, and current status) to determine if it correlates with the pilot's perceptions and understanding of the current situation. This information, if different, will be used to update the expert systems. At this time the pilot asks the a/c to perform a set of maneuvers, with and without the pilot's assistance. This allows the a/c to do an error analysis which will enable the pilot and the a/c expert system to know the working performance level (i.e., accuracy) of both. This will also increase the reliability of the confidence ratings for the "risk level" of specific maneuvers during the missions.

As the a/c approaches the target area, the pilot orders it to initiate the preprogrammed action to automatically take over certain controls that are necessary to keep the pilot's workload at a desired level during the attack run. During the attack, the a/c can sense the physiological changes of the pilot whenever he/she prepares to commence an overt action, responding without delay to the commands of the pilot and giving priority to any possible tactical actions that the pilot might attempt.

The a/c, while monitoring the environment, will warn the pilot visually, audibly, and/or tactually of any hazards or threats. Visually these hazards or threats will always be readily apparent to the pilot, since the a/c will integrate and display multiple sources of long-range sensor data directly on the pilot's retina in a hybrid pictorial/symbolic format. The presented data will require minimal reencoding or interpretation to be incorporated into the pilot's situation awareness. Top priority for the a/c will be the safety of the pilot. It will only deviate from course as long as necessary for pilot safety, with respect to completing the mission flight plan in accordance with the previous tactics. With every deviation from the planned mission, it will suggest new tactics/maneuvers and their probabilities of successful completion. The a/c will be able to assess the mission, and alert and data-link with other units to the extent that the a/c will function as a mini $C^3$ platform. The ability to bring into communication other units, having other weapons not carried by the a/c (e.g., long-range smart missiles), that are able to complete the mission or handle any novel situations while the a/c is occupied, will be invaluable.

### 3.3.3 Analysis of Pilot/Aircrew Automation Scenario

First, there are a few capabilities mentioned in this scenario that depend on achievable physiological/psychological monitoring technologies. Examples are sensing physiological changes in the pilot preparatory to an overt action and displaying information directly on the pilot's retina in a way requiring minimal reencoding by the pilot. The first, for instance, depends on developing adequate sensors and on psychologists/physiologists determining that there are patterns in the sensory data that reliably signal that an overt action is about to occur. Once these scientific developments occur, it is an AI problem to computationally recognize these patterns in the data. As a result of the dependence on psychological and physiological progress, we will not further discuss these.

The following capabilities are required to achieve the pilot/aircrew automation scenario:

- A tactics planner (para. 1, 3, and 5). Note that the system must reason about best tactics, success likelihood, etc. in the light of adversarial action. Furthermore, it must seek solutions in light of multiple goals, such as mission success and pilot safety.

  Coordinating diverse activities in real-time (para. 2 through 5). The activities include understanding various kinds of sensor data, controlling various devices (e.g., displays and aircraft configuration), speech processing, customizing presentation to the pilot, assessing the situation, and planning tactics.

- Situation assessment (para. 3 and 5).

- Speech understanding and synthesis in a limited domain (para. 1, 3, 4, and 5). (It is clear that speech is needed, since the pilot should not have to type input.) The domain here is relatively more limited than for the other two scenarios. In particular, the domain of discourse includes pilot commands, device readings, tactics and situation of the aircraft.

- Understanding ill-formed and noisy input (para. 1, 3, 4, and 5). The pilot may use ungrammatical or otherwise ill-formed speech in severely stressful situations.

Perhaps the most difficult problem will be coordinating the diverse activities in real-time. This is not merely a problem of computation speed. Rather it is fundamentally a problem of controlling the various activities, focusing attention on the important data/results, and distributing attention over the diverse activities. This is a problem that has been little addressed in AI.

A significant simplifying factor is that the scenario involves a nearly limited domain. For instance, the tactics planned need only know about strategy of its aircraft, enemy aircraft, SAMs, etc., but need not reason about more global issues.

## 3.4 COMMAND, CONTROL AND COMMUNICATION ($C^3$)

### 3.4.1 Functional Specifications

A. Communication

- The system is adaptive to workload/task demands to facilitate team member communication/coordination between units.

- The system utilizes natural language understanding and synthesis to communicate with operators.

- Mixed sensor and other electornic inputs are accepted by the system as alternative communication and for maintaining $C^3$ system network integrity.

- Smart attention cuing allows the system to communicate information to $C^3$ teams in any workload situation in a prioritized manner.

- Rapid turnaround of data to alleviate equipment-lage functions.

- The system tailors display interfaces to optimize information portrayal and format respective to the level of language required for any particular $C^3$ member.

- The system provides automated checklists to allow user interaction with the logic process during a $C^3$ situation.

B. Expert Understanding

- The system analyzes all data entered, identifies and extracts pertinent data, and computes a probability (%) of correctness for data transformation.

- The system is programmed to understand awareness: environment, defensive and offensive tactics, weapons systems and characteristics, roles and mission, knowledge of enemy and own characteristics, and operator experience and training.

- The system accepts and integrates new data, and does a data reliability assessment.

- The system is able to form a hypothesis of situational outcomes for planning.

    - The system uses historical data of past $C^3$ situations.

- The system copes with novel situations to maintain operational efficiency.

- The type of situation analysis which could be done by AI technology is constantly searched for by the system.

- The system understands the $C^3$ authority levels/hierarchy to enhance its interfacing with $C^3$ personnel.

- System reconfigurations (e.g., in an emergency) to ensure the overall $C^3$ system integrity.

C. Decision Aiding

- o Auditable probability assessments of tactical plans and/or situations are given as needed..
- o Speed/accuracy tradeoffs (probability) are available for uncertainty reduction.
- o Real time speed of data transfer for rapid turnaround of information.
- o Intelligent inferences of high quality are made from pertinent data.
- o An allocation of task and control of $C^3$ equipment and units, in $C^3$ situation, to the appropriate personnel are made by the system.

### 3.4.2 Command, Control, and Communication Scenario

In the northern Atlantic, a NATO E-3A airborne warning and control system (AWACS) aircraft (a/c) has discovered that a large number of various enemy attack a/c have left enemy controlled airspace and are heading for the United States and Europe. Fighter a/c have already been launched from the nearest NATO airbase to intercept. The AWACS is directing these a/c while simultaneously communicating with a ground air-command station which has linked up with a command center at the Pentagon. The commander of the Pentagon's command center asks the $C^3$ system to: (a) verify that all command centers in the network have been alerted and are activated; (b) establish a secure voice-link with all command centers yet to be briefed on the current situation; and (c) continually display current positional data of all forces to these command centers. Positional information of both friendly and enemy forces could be displayed pictorally or symbolically, and described to any unit's commander in plain English and/or military jargon.

The Pentagon $C^3$ system's expert knowledge base contains complete $C^3$ operations schemata of past tactical and/or battle situations that have been recorded from texts, $C^3$ experts, and training exercises. All participating $C^3$ units, peripheral units (i.e., field units such as a/c and sensor equipment), and personnel respective to these units with their chains-of-command are included in the $C^3$ system's expert knowledge base. The $C^3$ system can continually monitor the processed data from data-linked sensor equipment and situational inputs sent from its remote command centers so as to update, evaluate, and infer hypotheses of possible outcomes of various tactics.

With a combination of some human supervision and explanation capability from the expert data base, the system can decide which data are pertinent and need to be highlighted and/or further analyzed, and then separate these data from the total input of sensor and situational data. Computations for these data with regard to its probability of correctness are displayed and/or audibly read, as desired by the Pentagon commander. The system also has the ability to reconfigure its coverage and communication links with all units and/or sensor equipment in case of any new situational emergencies (e.g., the destruction or failure of a field unit).

In the $C^3$ system network, each $C^3$ center's ground personnel, and those of pilots and sensor operators in $C^3$ capable a/c will have their vital signs monitored, enabling each system to assess stress levels of the users for workload computations. In covert secure situations the a/c would not radiate/transmit any physiological data, and only if desired in other situations. These physiological computations provide priorities and considerations to be used by each $C^3$ system and peripheral units in allocating tasks and controls to various users. With knowledge of combined and individual unit battle tactics and the workload assessments of each unit, the $C^3$ system is able to decide which units and/or equipment are most important in any situation and then assign priority in communication/data-link for rapid turnaround of data. This close to real time data enables each $C^3$ system to make relatively high quality inferences, and to compute speed/accuracy tradeoff probabilities for uncertainty reductions to aid $C^3$ commanders in strategic decisions.

The commander of the Pentagon command center queries the $C^3$ system as to its prioritization of units or equipment. The commander may, if desired, change or prioritize differently from the $C^3$ system, or have any particular segment explained for clarification. Clarification could amount to simply an explanation of the logic used by the system in making its choices, or a firsthand look at any tactical display from remote units of sensor equipment on any a/c, ship, and/or ground station linked in the $C^3$ network. The commander also asks the system if any missile activity has been detected by the North American Defense (NORAD) combat operations center. The system responds that no missile activity has been detected, but as a precautionary measure NORAD has activated the missile defense system, and will be assuming command of the $C^3$ system network and that the Pentagon's $C^3$ commander is advised to be prepared to evacuate to the Strategic Air Command (SAC) E-4 a/c if missile activity is detected. The NORAD commander asks the system if any unidentified airborne contacts are being monitored. The system reports that unidentified airborne targets are being monitored on the originating E-3A AWACS.

A Captain working aboard the E-3A monitors 50 airborne targets. The Captain's system is "smart" enough to positively identify 12 targets as "HOSTILE," and 30 others as "FRIENDLY." The Captain concentrates his/her attention on the eight remaining targets designated as *"UNKNOWN." Fatigue* is beginning to set in, and the displays at the Captain's console change subtly. Recognizing that the Captain's reactions have slowed, the system begins to include highlighting and audible signals to help his/her performance. Some information previously displayed (viz., friendly contacts) disappears from the screen, letting the operator concentrate on the most important activities. At one point the Captain asks "What's the probability that target Bravo seven (B7) is hostile?" "Less than ten percent, based on known threat characteristics," responds the system. The Captain classifies contact #B7 as "FRIENDLY" and continues to monitor the scope. The a/c's $C^3$ system notifies the a/c's commander that the crew is getting fatigued. The a/c commander then passes this message to the NORAD $C^3$ system.

The NORAD system having acquired data on the fatigued state of the crew, signals the NORAD commander. In response, the commander tells the system to locate the closest airbase to the AWACS currently on-station, and alert the Ready-One Crew at the airbase to takeoff and relieve them. The system answers that the closest Ready-One AWACS crew will be on-station in 15 minutes, be briefed in flight through the secure data-link network, and the E-3A being relieved will exchange its data with the Ready-One E-3A en-route to the on-station area.

The system signals the NORAD commander that an encrypted message from the enemy has been intercepted by a field unit. The NORAD commander asks the $C^3$ system if the message has been sent to a decoding center. The system replies that it has, and that the deciphered message will be transmitted to the commander on a top priority basis. The commander begins to request a visual update on the known positions of both friendly and enemy forces, but is interrupted by an audio alert from the $C^3$ system. The intercepted message has been deciphered and is being transmitted to the NORAD commander along with a probability rating of its correctness. The message, with a .9 probability rating, relates that the enemy wants to have a nuclear class submarine release its payload from the northern Atlantic. The NORAD commander requests that the Admiral in charge of the Atlantic Fleet anti-submarine warfare (ASW) squadrons be immediately notified at an ASW $C^3$ center about this message, and to deploy any ASW forces near the northern Atlantic area. These ASW forces are to attempt to locate and stop the submarine from releasing its payload.

The Admiral, standing-by at the ASW $C^3$ center, acknowledges the receipt of the message and the order. The Admiral then selects and signals an ASW Squadron to launch their Ready-One (R-1) and Ready-Two (R-2) a/c. Due to time constraints, not only the R-2 crew will be briefed in flight by the previously briefed R-1 crew via a secure voice data-link, but both a/c will have positional data transmitted to them by the ASW $C^3$ system.

The two ASW a/c have arrived on station and are sweeping (i.e., scanning with) their radar and also monitoring all known radar parameters. A sensor station (SS) operator on the R-1 a/c has a SS scope saturated (i.e., covered) with electronic countermeasure (ECM) cuts (i.e., bearing lines received from other radars) such that the operator tells the system, "Only accept and display cuts from enemy submarines. Display cuts from the primary target (i.e., the nuclear sub) in red and any others in green with the respective radar parameters displayed on all cuts." The system clears the SS scope of the unqualified symbols and leaves two intersecting red bearing lines with the nuclear class submarine's radar parameters displayed at each cut. The SS operator instructs the system, "Pass a data-point of the red ECM intersection to the Tactical Coordinator's (TACO) scope." The SS operator also notifies the TACO that there was no visible radar contact at the ECM cut intersection.

The SS operator then queries the system, "What's the probability that the ECM symbols are from a false target, and if not, what's the probability that the submarine could have detected us and submerged?" "Target is 90 percent true, and there is a 90 percent probability of submerging from detection of our a/c," responds the system. The SS operator is shocked, and still believes that the target was false since there never was any visible radar contact. The SS operator asks the system, "How can the contact be 90 percent true?" The system replies that it had detected a small radar pulse return, indicating a possible radar target, at the same location as the data-point. This was during the first few sweeps of the radar when the operator's SS scope was saturated and the operator could not discern the contact. The SS operator then instructs the system to compute the time since the possible radar contact disappeared and display the data to himself/herself and the TACO.

The time displayed was 1.5 minutes, and the TACO decided that it was still possible to pinpoint the sub using the magnetic anomaly detection (MAD) system. The TACO tells the a/c $C^3$ system to display a fly-to-point (FTP) on the pilot's scope at the same location as the data-point. The pilot heads the a/c for the FTP to catch the submarine.

The tactical data on the ASW a/c is also being sent back to the ASW $C^3$ system and then onto the main $C^3$ system network. The commander at the NORAD $C^3$ system is signaled by the Admiral to note the current status of the ASW effort. The commander requests the $C^3$ system to state the current status concerning the ASW effort. The $C^3$ system at NORAD announces, "The enemy submarine has submerged without firing weapons, and is being kept under ASW surveillance." The E-3A AWACS, which relieved the originating AWACS, signals the NORAD commander that not only is there still no missile activity in their region, but that the enemy's a/c have been intercepted and are heading back into their own controlled airspace. The NORAD commander acknowledges the message and requests that the current status of the situation be relayed to the Pentagon $C^3$ system's commander.

The Pentagon's $C^3$ commander acknowledges the receipt of these data and notifies the NORAD commander that the Pentagon's $C^3$ center will resume command status. The Pentagon's commander also requests the NORAD center to stay active and continue to monitor the further surveillance measures by the NATO forces until any counterattack is deemed appropriate by the President.

### 3.4.3 Analysis of $C^3$ Scenario

As in the pilot/aircrew automation scenario, there are capabilities here that depend on advances in monitoring physiology/psychology, before AI systems could be developed to enable those capabilities. These include monitoring the vital signs of pilots and sensor operators to assess stress levels (para. 4 and 8). We will not address these.

The $C^3$ scenario presumes a number of capabilities.

- Natural language generation (para. 1, 11, and 13). Note that the level of the system is highly sophisticated, since it can brief a commander in language specific to his/her level of expertise. This assumes a model of user goals and expertise.
- Analogical reasoning based on stored schemata of analyzed situations (para. 2).
- Natural language understanding in a broad domain (para. 1, 3, and throughout). The input consists of both "situational inputs" and (possibly spoken) requests from $C^3$ commanders. The domain is

broad since the requests may include discussion of broad tactics, individual units, records of analyzed situations, and conditions of an individual.

- Speech understanding and synthesis in a broad domain (para. 1, 7, 10, and 11). Speech seems assumed throughout in fact for rapidity of interchange.

- Coordination of all processed data and conclusions there from (para. 2 and 3). The range information includes processed sensor data, natural language input, each individual's condition, tactical decisions, network communication status, and capabilities of all units. All is to be done essentially in real-time.

- Integration of displays and natural language to convey information (para. 4 and 6).

- Signal understanding (para. 7 and 10). The system recognizes friendly/unfriendly aircraft and their threat characteristics; this conveys understanding of the various sensor data.

The most difficult aspect of this scenario is that the expertise and language capabilities must range over a broad domain. All success thus far in AI has been based on narrow domains. As the number of facts of a domain increases, the number of alternatives to be considered in reasoning and in natural language processing grows far more rapidly. Consequently, techniques that work effectively in narrow domains may not be effective in broad domains.

Notice that the scenario is vague regarding whether the $C^3$ system is a single, monolithic system cognizant of all issues from the data to global strategy or a diverse collection of aids that alert the appropriate commander for his/her decision or action.

This vagueness is deliberate, since it is not possible to predict when, if ever, a single monolithic system would exist. What is technically feasible and what is desirable from human factors concerns in system design is unclear at present. Consequently, the exact division of "the system" into various aids to commanders will be unclear for some time.

## Section 4. Required Artificial Intelligence Capabilities

Walter Reitman, Ph.D.
Ralph Weischedel, Ph.D.

### 4.1 INTRODUCTION

Eight areas of AI research have been identified as critical not only to the capabilities evident in the scenarios, but also in general to military applications. These are:

- o expert systems
- o natural language processing
- o knowledge representation
- o computer vision
- o intelligent tutoring and training
- o planning
- o AI tools and environments
- o speech

Each of the eight AI areas is examined from the viewpoint of research needed to achieve the capabilities in the scenarios; milestones for achieving them are also provided here. An overall view is provided in section 4.10 via several tables summarizing the milestones and observations about the conditions fostering successful AI research.

Throughout this section the capabilities assumed by the scenarios are grouped by AI subareas; in parentheses after each capability is the list of scenarios in which it arose. Though we have categorized each of the capabilities very neatly into one of eight subareas of AI, these capabilities simply cannot be so neatly pigeonholed. These eight AI subareas have tended to function somewhat separately in the past; however, we expect these areas to interact much more substantively and frequently in the future. Certainly, these capabilities cannot be attained without such interaction between the areas.

### 4.1.1 On Defining Milestones

A milestone, to be useful, must clearly define a state of technology such that one can determine whether the milestone has been achieved. An estimate of when that milestone could be achieved, given a specific level of additional effort, is critical for planning. Any dependencies between milestones or other technologies must be specified, since failure to achieve projections regarding them will impact achieving the milestones depending on them. For each milestone, the information above is provided. Furthermore, where alternative technologies exist, these are identified.

### 4.1.2 On Quantifying Projections

There are several difficulties due to the nature of AI that make quantifying any projections highly unreliable.

- o AI research is, in a sense, in its early childhood, and AI applications are in their early infancy. It is generally agreed that the first AI endeavors were in the late 50's. Commercial products resulting from AI efforts have become available in the last five years.

- o There is little knowledge of the limits of the science. In physics, the discovery of natural laws such as the laws of thermodynamics and of electromagnetism indicate that certain goals are impossible, e.g., a perpetual motion machine or movement faster than the speed of light. Perhaps the only comparable result in computer science is that some problems are unsolvable; unfortunately the mathematical problems that have been proven unsolvable bear little relation to problems in simulating intelligent behavior.

- o The scenarios are designed to be at the endpoints of what could be possible, rather than short-term extrapolations from current technology. Unfortunately, it is not clear whether AI is near strategic breakthroughs that would make possible significant leaps forward.

- o Successful AI research, and almost all effort, has been in projects where the AI system deals with a single, narrow domain, e.g., natural language access to a single data base, or an expert system for glaucoma treatment. The scenarios generally involve broad domains and/or interactions among several domains.

- o Hardware advances, while they have made AI applications more affordable and have brought more effective programming environments, have not led, in general, to solving unsolved AI problems. Rather, scientific insights (and secondarily the software bottleneck) remain the key to advances in AI.

Nevertheless, it is already clear that useful AI applications are becoming available. Though quantifying when a given milestone will occur and how much effort it will take to achieve it is highly unreliable if not impossible, identifying those milestones and the logical dependencies among them is reliable and critical.

Consequently, we make recommendations for additional research efforts, but cannot predict with any assurance that that will lead to achieving the scenarios or intermediate milestones. Increased effort in the amounts projected (FY-84 dollars) will, however, definitely lead to far more effective systems that partially achieve the specified capabilities.

## 4.2 Expert Systems

### 4.2.1 Capabilities Required by the Scenarios

A. Advice on relevant (highly diverse) data bases and individuals (SYSTEMS DESIGN). This is a very complex capability in that it requires understanding of not only the content of the data base, but also the goals of the user and the strengths and weaknesses in expertise of the user. The same goes in advising regarding relevant individuals. One needs to model the expertise of the individuals in order to advise as to who may most appropriately help. As a consequence, this is not simply an expert systems capability, but also depends very heavily on knowledge representation progress.

B. Model of user goals (SYSTEMS DESIGN). The need of user goals is clear from the first capability as an example to advise regarding relevant data bases. To know what is relevant, one certainly needs to know the goals of the individual user. Furthermore, it's also critical to the natural language dialogue that is occurring between the individual user and the expert system. The reason for this is that the user is communicating very succinctly the requests they have in mind, but the way that the dialogue is kept so succinct and natural is that the expert system actually has a model of the underlying user goals. Therefore, requests can be made briefly without having to spell out every last detail. This capability of modeling of user goals is not only important to the expert system area, but is also important to natural language processing. Furthermore, it depends on the knowledge representation research as well. Thus far, consideration of user goals has primarily been a topic in natural language research.

C. Model of user expertise (SYSTEMS DESIGN). This capability is also needed just as the model of user goals to allow for individualized advising of the user and also to allow the natural language capability. It's very important in both issues so that in advising the individual regarding his/her needs one is making recommendations appropriate to the level of help he/she needs. One should not give obviously low-level information to someone who already knows that information; by the same token, providing advice at a level beyond the understanding level of the individual is not providing advice in any useful sense. This impacts the natural language generation capability for the same reasons. One needs a model of the user expertise so that one uses terms appropriate to the individual. This point therefore, is a capability that interacts with not only the expert systems subject area but also natural language processing and knowledge representation.

D. Model of changes in cockpit design impacting simulator design (SYSTEMS DESIGN). This clearly depends substantially on the underlying knowledge representation, so that one can represent various kinds of information about the cockpit design, such as spatial layout, such as component design. To represent the impacts on simulator design, there has to be a very robust reasoning component and also a rich model of causality. Causality is involved since one needs to be able to explain why a particular cockpit change affects the simulator design.

E. Reasoning (including probabilistic reasoning) about fidelity of simulator (SYSTEMS DESIGN). At one level, this is very similar in nature to the previous capability, in that one has to be able to reason about various properties of the cockpit design and how close the simulator relates to the typical movements and activities that a pilot uses given that cockpit design. Consequently, the same issues in knowledge representation and reasoning arise, but in addition, there is the need to provide numerical probabilities regarding the fidelity.

F. Writing typical contracts (SYSTEMS DESIGN). This clearly depends on a good model of what is typical and how the current situation differs from what is typical. Therefore, a kind of common sense reasoning regarding typical contracts and the current situation is needed. We will argue later that in fact common sense reasoning capability is the critical issue here, and that this is more an issue of knowledge representation than natural language processing. In fact, it is possible that in this situation natural language generation would be inappropriate, since one normally wants to use standard clauses and standard variations on standard clauses in writing contracts rather than writing new clauses from scratch.

G. Real-time situation assessment (PILOT/AIRCREW AUTOMATION, $C^3$). This capability is not only a problem in expert systems, but also a problem in knowledge representation and reasoning. This is because one critical aspect of situation assessment is going to be drawing analogies with past situations that have been recorded in some data base.

### 4.2.2 Current and Near-term State-of-the-art

First, let us consider what technology is present now or likely to be present within the next five years, given anticipated funded research. Naturally, very little of that is specific to the capabilities identified in the scenarios; rather it is primarily aimed at advancing the technology, in general, e.g., aids to knowledge engineering.

The first generation of expert systems tool kits is now available. The best example of such a system is the KEE System, produced by IntelliCorp, and available for about $60,000. Others include the Loops Language from Xerox, ART from Inference Corp. and the Expert Tool Kit available from Rutgers University. These are provided for the programmer; as such, they provide some aid to knowledge engineering by making it easier for a programmer to codify his/her understanding of the expert's knowledge and reasoning.

In two to four years we should see the next generation of systems, where the existing system tool kit capabilities will be integrated with other program components, and will employ both richer representation languages and more varied control structures. Sources of these capabilities will be the private AI firms and AI labs involved in the Strategic Computing initiative.

Due to the high investment in research in expert systems arising from venture capital companies and industrial labs, current technology will be advanced in:

- o diversity of applications
- o software tools for constructing expert systems
- o some aids to acquiring knowledge

The DARPA Strategic Computing effort will also contribute substantially to the next (second) generation of expert systems. In particular, it will contribute uniquely to bring the results of exploratory research to a level of maturity such that it is ready to be applied. Of the areas addressed, one should note particularly:

- o increased speed (most expert systems do not run in real-time)
- o advances in automating the acquisition of knowledge from experts
- o demonstration of military applications (e.g., Pilot's Assistant)
- o extension of expert systems to domains where human expertise does not exist

Since knowledge engineering is such a critical aspect of building expert systems and since knowledge acquisition is a key problem in the knowledge engineering process, some discussion of the prospects for and limitations of aids to knowledge acquisition is appropriate here. There are three approaches one could envision in aiding knowledge acquisition.

1. One could provide better tools for the programmer to codify information systematically with greater guidance for the programmer or knowledge engineer in what information to elicit and how to codify it.

2. One could develop very high level programming languages, such that it is much easier for the expert to encode his or her knowledge directly into the programming environment, and yet still not demand the kind of detail inherent in current programming processes. Of course, this still would assume that the expert does programming, though at a very high level compared to current programming requirements.

3. One could envision an automated system which plays the role of the knowledge engineer or programmer. Namely, the expert types in English the knowledge and reasoning used, and it is automatically encoded into appropriate programs. We consider this prospect well beyond the foreseeable future, and therefore well beyond what anyone can predict or plan for.

Both of the first two prospects are likely to succeed at least to the degree that we will see enhanced tools along both paradigms. Of the research activities outlined in this section, the third approach is the only one with substantial limitations.

Since the goals of the second generation do not, in general, involve problematic research issues, the probability of the research efforts impacting technology and applications is high, probably .9 over the next four years. The area where problematic issues may surface is in applying the technology to domains where no human expertise exists, since previous applications have depended on the existence of an expert that can introspect about their decision-making.

Preparing for the "third" generation of expert systems is where significant additional funding is necessary. Some fundamental characteristics of that third generation are stated as milestones; the areas that need support and the level of support needed for achieving those are stated there.

The third generation depends so fundamentally on projected fundamental results in knowledge representation and natural language that probability estimates would be arbitrary.

What characteristics are desirable in systems after the second generation, in order to make progress toward the milestones? Here we repeat the expert system capabilities listed in section 4.2.1 and assumed by the scenarios.

1. Advice on relevant (highly diverse) data bases and individuals (SYSTEMS DESIGN).

2. A model of user goals (SYSTEMS DESIGN).

3. A model of user expertise (SYSTEMS DESIGN).

4. A model of changes in cockpit design impacting simulator design (SYSTEMS DESIGN).

5. Reasoning (including probabilistic reasoning) about fidelity of a simulator (SYSTEMS DESIGN).

6. Writing typical contracts (SYSTEMS DESIGN).

7. Real-time situation assessment (PILOT/AIRCREW AUTOMATION, $C^3$).

Items two and four above fundamentally require knowledge representation advances to represent goals, belief, space, and causality. This is discussed in the sections 4.4 and 4.7 on Knowledge Representation and on Planning.

Item one assumes both the ability to reason about (highly diverse) data bases or sources of knowledge and also ability to verbalize this to someone based on their level of expertise. These two abilities are covered separately in the first two milestones below. Item three has been studied in natural language efforts and is therefore treated with the milestones of that section.

Item six, writing typical contracts, might at first seem to require natural language processing. However, writing truly typical contracts may only require knowing which standard clauses to include, which to exclude, which standard modifications to include, and where standard blanks must be filled by the appropriate names, addresses, dates, etc. Thus, it does not necessarily involve natural language processing. Rather, it seems to involve more straightforward reasoning about each clause of a typical contract.

The major missing requirement for reasoning about fidelity of a simulator (item five) seems to be analogical reasoning, which is covered in section 4.4 on Knowledge Representation. The last milestone in section 3.2.3 deals with situation assessment, item seven.

### 4.2.3 Milestones

#### 4.2.3.1 Explanation

An expert system must be able to respond to any question about what it knows, why it did something, whether it considered something, why it didn't consider something, etc. Otherwise, there is inadequate basis for understanding its recommendation, for knowing whether it took into account all criteria important to the user, and therefore for developing trust in its recommendation.

We expect this event to take place within 15 years. This milestone assumes:

- a relatively complete natural language understanding capability, including the ability to understand the user's intentions, goals, and beliefs

- high quality natural language generation to enable accurate, understandable explanations including: omission of material the system believes the individual user knows and summaries of chains of reasoning the way an expert would

Current technology allows only simple explanations based on simple knowledge structures, e.g., <u>if</u> A & B & C <u>then</u> D. The "explanation" is merely an English gloss of such a low-level rule. It is clear that that is inadequate as explanation. There are also domain-specific explanatory capabilities, which may not transfer to an expert system for another domain. Some advocates of the techniques described in this paragraph deemphasize the need for a system to be capable of general understanding of questions and the intent of questions in order to be able to provide pertinent explanations.

However, the scenarios could not be achieved with current technology; nor is it likely that any extension of those techniques could achieve anything like that technology.

Since the milestone involves natural language generation at its heart, the best way to approach achieving it is through natural language research whose domain is expert systems. (See the Natural Language section for level of effort estimates.) A substantial deficiency in current research efforts is viewing expert systems, natural language processing and knowledge representation as separate topics.

#### 4.2.3.2 Integrated knowledge sources, including reasoning across domains

This includes several aspects:

- o Integrating expertise from various mathematical bases, e.g., symbolic reasoning, numerical simulation, statistical inference, and computer algebra
- o Integrating reasoning from differing knowledge sources in one domain, e.g., sensor data, tactical knowledge, and symbolic input
- o Integrating reasoning across domains, e.g., human factors data, engineering specifications, and tactics data bases in the systems design scenario

Both items one and two arise as necessary characteristics of the expert systems in the pilot/aircrew automation scenario and also in the $C^3$ scenario. This is because there are various kinds of inputs in both environments, such as sensor data, symbolic encodings of data bases of past tactics, symbolic input regarding the mission task, and symbolic data about likely enemy activities. Furthermore, the reasoning will involve several types. There will be reasoning best represented by equations, such as trajectories of aircraft. There will be probabilistic reasoning about likely behavior and likely success. In addition, there will be symbolic reasoning as a pilot might do about typical strategies and tactics to follow in a mission.

The third aspect listed above arises most obviously in the systems design scenario in that the designer has various data bases to access. The same problems arise, however, also in the $C^3$ environment, since the natural language input and output encompass several different domains, such as tactics, sensor data, positional information, missions, personnel, and hardware.

The first two items above tend to involve software integration issues more than departing from a narrowly defined domain. The third departs most from current capabilities and therefore is probably 10-15 years away, since it departs from the significant limitation of current technology, namely, representing knowledge and decision-making in a narrowly defined domain. The third is also least likely to be funded given current funding for applications. Consequently, we recommend additional effort of 3 person-years per year on this third problem; it should be at least doubled after five years. The initial amount is recommended since this much at least is required to fund two very small efforts or one modest-sized group to investigate these problems in a few small domains. Funding of at least 6-7 person-years per year after the first five years is needed to fund two small groups or one medium-sized group.

Integrating knowledge sources is quite an advanced question which is just becoming of interest to members of the community; there are no well-differentiated schools of thought with respect to it.

To attain the third goal of this milestone, several factors are important:

- experimental efforts to develop several different task types, e.g., diagnosis, analysis, and design, operating within the same domain. There are several issues. What scenarios would be most helpful for the user, for instance, a designer as in the systems design scenario? What interface allows effective communication for the intended user community? This almost certainly means a mixture of natural language and graphics; however, that does not imply much about the nature of the interaction nor the functional capabilities. In addition, what system-specific issues arise from answers to the first two questions? For instance, under what conditions should a component of one task type call another; what information, including resource limitations, should be transmitted; how can parallel processing be used among the task types and their components?

o setting up target domains in which these problems can be explored. The choice of target domains is critical to effective progress. The domains must be simple enough to minimize system-building problems for each domain taken separately, but must be just rich enough to motivate solutions to problems about domain interactions.

o locating points of correspondence and deviation between related frameworks. The points where the domains overlap are critical. For, commonalities and deviations are precisely where the system must show special expertise. It must be able to identify such connections for the user, who may not see them. It must be able to distinguish near connections and differences among usages of common terms, so that the user may receive clarification regarding potentially confusing terms. The amount of detail regarding field names, encodings of field values, etc. in an artificial language is bad enough for a user in a single domain. Reasoning across overlapping domains seems to require natural language interfaces; if the AI system can keep track of the terminological commonalities and differences, terminology will not be an overwhelming burden to the user, as it likely would be with an artificial language.

o knowing how to help users understand how to use sources of information that may be relevant, but which they are not familiar with, nor know how to incorporate into their usual task procedures. To provide such advising and help is itself quite a substantial reasoning problem.

#### 4.2.3.3 Planning in a rapidly changing environment with adversaries

It seems likely that, no matter how advanced our systems become, there will always be uncertainty, e.g., with respect to the adversary's plan and due to the adversary's attempts to conceal information. Consequently, it appears that research in the following knowledge representation and planning areas is essential:

o representing uncertainty

o expressing reasonable cases for making assumptions

o retracting conclusions as assumptions are invalidated

A system that does this in assessing tactics is perhaps as much as 15-18 years away, since current planners assume a single agent without adversaries or spontaneous changes resulting from natural events. Section 4.7 on Planning contains milestones for this.

## 4.3 NATURAL LANGUAGE PROCESSING

### 4.3.1 Capabilities Required by the Scenarios

A. Natural language understanding in a broad domain (SYSTEMS DESIGN, $C^3$). The key issue in this capability is the broad domain. All success with natural language thus far has been in narrowly defined domains.

B. Precise generation in a broad domain (SYSTEMS DESIGN, $C^3$). The problem here again is using natural language in a broad domain, for all success thus far has been in narrowly defined domains.

C. Understanding of ill-formed input as the user intended (PILOT/AIRCREW AUTOMATION).[1] This capability is very demanding because constraints such as grammar, vocabulary, and a narrowly defined set of facts about a domain are critical to understanding what is intended. Therefore, violation of any of those constraints makes understanding much harder for machines. Nevertheless, it is clear that people neither speak nor write perfectly. Rather, case studies show that errors of one form or another, even in written text typed to a computer data base, occur in as much as 25% of the queries to the data base.

D. Customized natural language generation (SYSTEMS DESIGN, $C^3$). This capability is clearly central to successful natural language generation. In communicating with an individual, it is clear that even things that we take for granted, like describing an entity that we want to refer to, are customized to the individual that we are communicating with. Therefore, this is a capability fundamental to all natural language generation.

---

[1] Ill-formed input, in fact, is likely to be a critical problem in all of the scenarios. Though the systems design and $C^3$ scenarios give examples of only polished prose, there is abundant evidence that people neither speak nor type that way.

E. Understanding based on a model of user goals and plans (SYSTEMS DESIGN, $C^3$). Evidence indicates that the reason natural language is so easy to use and convenient is in fact that not everything has to be spelled out. Rather, one can succinctly indicate what they need to an individual who already has some understanding of your needs and goals. This problem depends on knowledge representation as well to represent plans and goals of a user.

F. Integration of displays with natural language to convey information ($C^3$). One of the beauties of natural language is the fact that it lets you convey information not only exclusively in that language, but also in other means, such as charts, graphs, tables, and pointing.

### 4.3.2 Current and Near-term State-of-the-art

Today, there are some (pseudo-) natural language systems, but all are severely restricted. Substantially richer natural language systems, still without much pragmatics, will be available by late 1985. The second generation of natural language understanding systems having some pragmatic capabilities are not likely to be available until 1988 or 1989. There are no natural language generation products at present.

Artificial Intelligence Corporation, Bolt Beranek and Newman Inc., Symantec, and possibly some new companies will be offering natural language understanders for accessing data bases by the end of 1986. Research laboratories, particularly those supported by the DARPA Strategic Computing initiative, will also be sources of the second generation systems.

Since the effort to achieve robust, second generation systems with limited, but quite useful, capabilities requires no fundamental breakthroughs and requires primarily applied research within a well-defined framework, we judge the probability of success in developing them to be .90. Substantial success, sufficient to make systems far more usable than the second generation is also highly likely (e.g., .8), assuming increased funding as recommended in the next section. The basis of this projection is that a firm foundation for the research has al eady been laid.

Funding for this second generation should be adequate given corporate funding and the DARPA Strategic Computing initiative. What is most needed is funding now on problems critical to making systems beyond the second generation available in a timely way.

The capabilities assumed by the scenarios are repeated here, additionally broken down by topics logically antecedent to completion of each given capability.

1. Understanding based on a model of user goals and plans (SYSTEMS DESIGN, $C^3$). This is necessary for succinct communication without the burden of having to spell out every detail. The communication in both the system design and $C^3$ scenario is both succinct and effective.

   - models of user goals, plans, and preferences. Such models should represent the over-arching and low-level details of user needs.

   - recognition of user intention in well-formed input. If intention is not recognized, only literal understanding (and its consequent misunderstanding) is possible. Recognizing intention in well-formed (i.e., perfectly correct) input simplifies the problem.

   - models of the dialogue environment. This includes additional factors, such as the entities that may be referred to and knowledge common to the dialogue participants.

   - heuristics for understanding anaphora, deixis, and ellipsis based on realistic models of dialogue environment. (These are defined in the section reviewing the state-of-the-art in natural language). These occur continually in natural language, and are therefore critical to understanding.

2. Understanding ill-formed input as the user intended (PILOT/AIRCREW AUTOMATION).[2] Input may appear ill-formed due to grammar errors,

---

[2]Ill-formed input, in fact, is likely to be a critical problem in all of the scenarios. Though the systems design and $C^3$ scenarios give examples of only polished prose, there is abundant evidence that people neither speak nor type that way.

mispronunciation, faults in the communication medium, and lack of complete knowledge by the system. Case studies have shown it to occur in as much as 25% of typed communications. It also occurs frequently in oral communications.

3. Integration of displays with natural language to convey information ($C^3$). This arises in describing positional information conveyed via map displays and open for discussion. It also arises in discussing radar displays.

4. Customized natural language generation ($C^3$). Since the language generated in the $C^3$ scenario is oriented to many individuals of differing background, expertise, rank, etc., choice of vocabulary and level of description for individuals is important to effective communications.

   o Models of user expertise. The purpose is to avoid stating the obvious to different users.

5. Natural language understanding and generation in a broad domain (SYSTEMS DESIGN, $C^3$). Many different domains of expertise are discussed in both scenarios. In the systems design scenario, this corresponds to several data bases: engineering, human factors, personnel, tactics, etc. In the $C^3$ scenario, the need arises from diverse domains of discussion, e.g., sensor data, personnel, tactics, etc.

   o Semantics of vague terms (e.g., low) and vague quantifiers (e.g., few).

   o Variation in semantics of words and phrases in a broad domain (e.g., the same words may have widely differing meanings in various subdomains).

   o Axiomatization of mundane knowledge (e.g., engineering change proposals [ECP's] can affect a simulator, ECP's have a date, the designer may wish to be notified of new ECP's after simulator design has begun,... .).

   o Heuristics for generating natural text (e.g., sequences of paragraphs).

   o Recognition of and appropriate response to user misconceptions (e.g., a presupposition of what a user indicates may be incorrect based on system knowledge. Without correction, this can lead to miscommunication.).

   o Generation of clarifying questions and paraphrases of user input. This is essential for understanding user requests that are initially unclear or too vague.

- Parallel capabilities in both generation and understanding (so that it can understand what it can generate). Not having this feature is bound to lead to confusion for the user, since the terminology and forms a system generates would be assumed to be understandable to the system.

Items 1 and 2 above are covered in the second milestone in understanding user intention. Items 3 and 4 are an integral part of the third milestone on unifying generation and understanding. Item 5 is discussed in the final milestone.

There are two significant limitations to keep in mind. The first is that all successful natural language research and all successful research in reasoning up until this time have assumed that the knowledge and reasoning underlying the system are confined to a single, narrowly defined domain. When this assumption is removed, it is not clear whether heuristics that function acceptably in a single, narrowly-defined domain will continue to do so in a broad domain. The reason is that the size of the domain and the number of facts to be recorded in the knowledge base, if kept small, is a limiting factor to the number of alternatives that any heuristic must consider and/or eliminate. With broad domains, the number of alternatives may grow exponentially. The second problem or limitation is the amount of effort it requires to encode knowledge, vocabulary information, and the formal relation between terminology in the domain and its formal representation in the knowledge base are very programmer intensive. Therefore, building natural language systems suffers from the same limitation that building expert systems does. Namely, the effort in encoding sufficient information to make a natural language system or an expert system work is a long-term problem requiring programming effort to build or extend these systems.

### 4.3.3 Milestones

#### 4.3.3.1 A well-scoped, practical domain

Most natural language understanding systems so far have utilized extremely narrow and restricted domains and simplified problems still further by imposing a formal model of the domain on the user.

The milestone is achieved if the system can carry on extended dialogue about a practical problem in a well-scoped target domain with competence near a human's. This requires limited work on several of the problems listed above, but simplified substantially by dealing with only a single domain, by not having a rich model of pragmatics, and by not requiring equivalent capabilities in generation and understanding. This milestone should result from current funding sources, in particular from the Strategic Computing initiative.

#### 4.3.3.2 Understanding user intention

This milestone is for a system that understands user intentions whether input is ill-formed or well-formed. It requires work on:

- knowledge representation of beliefs, goals, plans, and preferences
- adequate models of dialogue environment
- detection of shift in goals during dialogue
- detection of some user misconceptions
- development of recovery strategies for classes of ill-formedness
- empirical study of effectiveness of strategies
- ellipsis
- anaphora and deixis

Critical to the success of this research is selecting a problem-solving domain with sufficiently rich dialogue environments; a variety of user goals, plans for achieving them, and dialogue goals (such as clarification and explanation). This could be achieved in some expert advisement environments or problem-solving environments that involve at least one data base as a tool. The domain should provide an ample realistic corpus (collection) of dialogues representing how the user and expert (or system) should interact.

Each of the areas in the list must be advanced from ideas investigated in Ph.D. dissertations to the level of laboratory-tested results. Preferably, the individuals should be working at one or at most two sites so that hypotheses, programming, and discussion of these highly interrelated topics can advance most effectively. This means a level of support of at least 5 person-years per year. Given this support, laboratory demonstration of this milestone in a single domain should be possible in four years, and across domains, such as in the systems design and $C^3$ scenarios, in eight years.

#### 4.3.3.3 Unifying generation and understanding

The goal here is to enable systems to understand what they can say and vice versa. Current systems use essentially disjoint knowledge and processing for the two capabilities. A first step is to take some existing large grammars either for generation, such as NIGEL from USC/ISI, or for understanding, such as RUS from BBN and produce a comparable grammar which can be used for both understanding and generation. This can be done in 3-4 years, since substantial grammars such as NIGEL and RUS exist already, and since formalisms appropriate to the need are on the drawing board. Given that starting point, an effort of 3 person-years per year should be sufficient. Part of the task will be selecting an appropriate representation and algorithms. This should produce a grammar distributable for use in many other research efforts.

Another step is specifying use of ellipsis, anaphora, speech acts, and deixis in a way usable by both generation and understanding. This will advance most effectively after two years' progress and a common grammar and in modeling user intention and dialogue environment. Additional funding for 3-4 person-years per year starting about two years from now is probably necessary for this purpose. Four years from now, after the large grammar is available, an effort of at least 6-7 person-years per year is needed. The basis for this estimate is that adding these features is as hard as building a broad grammar.

This milestone should be achievable 6-8 years from now as a demonstration. Though the grammar itself should be available at the midpoint of this time, results from the milestone on user intention must be integrated into the understanding phase, plus making the processes for understanding intention available for conveying intention in generation.

#### 4.3.3.4 Explaining and paraphrasing

As section 4.2 on Expert Systems argues, a crucial element in using future expert systems will be the capability of explaining why certain decisions were made, describing why other recommendations were not offered, and paraphrasing the system's understanding of user requests and problems. Work on precisely paraphrasing system understanding, for instance, to clarify and verify what was meant should start immediately at a level of 2 person-years per year for the first two years. This provides for a very small project to explore strategies specific to clarification issues.

As work in knowledge representation enables recording dependencies in reasoning, research in explanation can progress. Most importantly, explanation work should go hand-in-hand with a simplified expert system prototype employing nonmonotonic reasoning, belief revision, and the ability to view a chain of reasoning steps as a single, well-motivated decision based on itemized premises and evidence.

After the first two years, we recommend additional funding of 5 person-years per year. This provides for one or two modest-sized efforts involving at least one person from knowledge representation, one from natural language, and one from expert systems. Useful capabilities should arise within eight years. The milestone is relatively far away since work in generation and in recording dependencies is relatively new, and since there has been even less work in explanation.

To additionally bring such capabilities to more realistically complex expert systems, an effort of at least 10 person-years per year is needed starting after eight years and continuing for an additional seven years. This will address issues of summarizing chains of reasoning, explanations of why alternatives were rejected, and adequate explanations involving a mixture of symbolic reasoning and simulation.

#### 4.3.3.5 Natural language in a broad domain or across narrowly-scoped domains

The next major milestone is to provide natural language processing across domains rather than in a single narrowly scoped domain. This presumes that substantial progress has been made on the previous milestone. The level of natural language achieved in this milestone is also greater than that in the previous one. Rather than first demonstrations in a single domain, the system should exhibit the result of extensive empirical work and hands-on experience with end-users to shape revisions in the techniques for:

- effectively using the speaker's goals, plans, and beliefs in understanding both well-formed and ill-formed input
- generating concise responses, explanations, and briefs based on the speaker's goals, plans, and beliefs
- understanding everything it can generate
- engaging in clarification dialogue and paraphrasing system understanding of the dialogue environment

This milestone provides the time for the theoretical frameworks implemented in the previous milestones to be hardened through experience and to be extended for overlapping domains. To achieve this milestone, substantial advances are needed in almost all of the problem areas listed in section 4.3.2. Choice of the domains for the first efforts is critical as well. In particular, one needs domains where confusion in terminology is possible, yet where the representation problems are tractable. Progress in knowledge representation is presumed in order to represent user expertise, goals, and beliefs and also to facilitate nonmonotonic reasoning: see section 4.4 on Knowledge Representation.

This milestone is 15-20 years away. This is based on the fact that all successful efforts to date in natural language, as well as in reasoning, have been based on confining the system to a single, narrowly-defined domain. Heuristics that work within that constraint may fail in a broad domain where the number of alternatives the heuristic must consider or eliminate may grow exponentially. Undoubtedly there will be some support for all of the problems mentioned in section 4.3.2. However, with significant additional support on those fundamental issues, the milestone could be achieved substantially sooner, say in 12 years. Since this milestone is so far out on the horizon, and since this deals with a problem that no AI research has tackled, estimates of funding required have no parallel. Support for the equivalent of at least three person-years per year initially, and at least two teams of ten person-years per year after the other milestones have been achieved, seems necessary.

## 4.4 KNOWLEDGE REPRESENTATION

### 4.4.1 Capabilities Required by the Scenarios

A. Model of user goals (SYSTEMS DESIGN, $C^3$). This is of course repeated from the expert systems capability and also contributes to natural language processing capabilities, as indicated above.

B. Spatial layout of cockpit to reason about bodily movement (SYSTEMS DESIGN). The problem here is that there is not yet a good model of space, such that computer systems can reason about space vaguely in a common sense way, as well as very precisely and numerically as in calculus. Therefore, this is a knowledge representation problem.

C. Data base contents and purposes (SYSTEMS DESIGN). This is of course repeated from the section on Expert Systems because one needs to represent the data base contents and the purposes of those contents adequately in order to provide advice regarding relevant data to an individual's needs.

D. Analogical reasoning, about designs and plans (SYSTEMS DESIGN), or about tactical situations ($C^3$). Analogy, while a fundamental aspect of common sense reasoning in humans, has been very difficult to model computationally. Nevertheless, analogical reasoning forms a basis in several of the capabilities mentioned elsewhere. Those capabilities all assume that some prior experience is encoded in the data base and the current situation or current needs are compared against that set of experience to indicate that which is most relevant in the past experience to the current situation or needs.

### 4.4.2 Current and Near-term State-of-the-art

There are a number of knowledge representation languages and processors now available though not necessarily as products. These include LOOPS (Xerox PARC), UNITS (Intellicorp), ROSIE (Rand), KL-TWO (BBN), KRYPTON (Fairchild), and SNEPS (SUNY Buffalo). All assume programming expertise. Within the next five years, languages like these will become more broadly used, clearer semantically, better integrated into programming environments, and more general in scope and control. This can be expected due to current funding patterns with probability .8 to .9, since each of these predictions involves incremental improvements on existing, much used systems. These developments will support applications of expert systems and natural language processors well during the next five years.

A number of needs are repeated here from the analysis of the scenarios and the subsections on Expert Systems (4.2.3), Natural Language (4.3.3), and Planning (4.7.3).

1. Representing space and time to support common sense reasoning, as distinct from numerical calculations (SYSTEMS DESIGN). In the systems design scenario, as the designer watches the display looking for choke points, the designer will reason about the spatial movements and their time of occurrence informally or symbolically. For instance, he or she is likely to comment that a particular switch is too near another switch without specifying precisely mathematically the notion of "too near." In a similar vein, the designer may reason and comment that a sequence of bodily movements given the current layout requires too much time without giving a precise number to how much time is too much. The symbolic representation of space and time must at least be able to support

reasoning about mutable objects, events, and actions.

2. Nonmonotonic reasoning and inexact knowledge (SYSTEMS DESIGN, PILOT/AIRCREW AUTOMATION, $C^3$). Nonmonotonic reasoning is a critical part of expert system and planning capabilities in all three scenarios. The reason is that in all three, there is a need to make reasonable assumptions when information is lacking due to adversarial covert action or due to a simple lack of information. Those assumptions are based on knowledge about what is a reasonable default assumption and drawing conclusions based on that assumption until some contradiction arises that invalidates those assumptions. If a discrepancy does arise, then the reasoning agent must know what conclusions to retract based on erroneous assumptions. The following subtopics are parts of nonmonotonic reasoning:

   o expressing the conditions on making reasonable assumptions when complete information is unavailable

   o recording dependencies in drawing conclusions (Such "dependencies" are the assumptions made in drawing a given conclusion.)

   o retracting previous conclusions when assumptions are no longer valid (The previous three together are called "nonmonotonic reasoning.")

   o representing <u>inexact</u> knowledge (e.g., "Few x's are y's.", "Dr. Smith rarely advises on simulator design.")

   o dealing with situations where several defaults may apply, where these lead to different conclusions

   o designing and representing strategies for dealing with discrepancies (Just throwing out assumptions is an insufficient strategy in the long run.)

3. Models of goals, beliefs, and preferences (SYSTEMS DESIGN, $C^3$). These are critical to the natural language aspects of the systems design and $C^3$ scenarios.

4. Models of data base contents and purposes (SYSTEMS DESIGN). The need for this in the systems design scenario is particularly evident since there will be several data bases from varying backgrounds and different kinds of expertise. For instance, the designer will have access to engineering specifications both of the current design and of past designs, human factors data, simulator design data, records of personnel available to help, etc. A model of the content and purpose of these varying data bases is needed to advise the designer about what is relevant to his or her needs and about what classes of

information are available from each data base.

5. Consciously controlling expenditures of computational resources on reasoning ($C^3$). Humans reason about how long they should spend on a given problem, how long they should take a particular approach, and how long they should plan. As we try to apply computer reasoning capabilities to larger and larger domains, an increasing problem will be the number of alternative possible solutions to consider in solving a problem or in drawing a conclusion. Therefore, it will become more and more important to specifically reason about how long a particular approach to a solution should be investigated, when to look for an alternative approach, and when to give up on a problem altogether.

6. Reasoning by analogy (SYSTEMS DESIGN, $C^3$). Such analogies are at the heart of basing current decisions and situation assessment on past, recorded situations.

All of these are likely to receive some funding. The problems of representing space and time; nonmonotonic reasoning; consciously controlling effort in planning; representing goals, beliefs, and preferences; and analogical reasoning are most difficult. Since there are substantial gaps that are not likely to be adequately funded, they will stymie advances in expert systems and natural language after the next five years.

Items one, two, and five are covered in the milestone on a common sense reasoning system in section 4.7 on Planning. Item three is part of the milestone on understanding user intention in section 4.3 on Natural Language. Items four and six are covered in the two milestones in the next section.

### 4.4.3 Milestones

#### 4.4.3.1 Data Base Content and Purposes

This milestone arises from the capability in the designer scenario to reason about the appropriateness of diverse data bases from different areas of expertise (such as human factors, training, and engineering specifications of cockpit designs) for a given individual's current problem. Nevertheless, it is far more generally useful since answering questions about the meaning of files of data and fields in them is a need arising in many data bases. There is general agreement regarding the need for such help facilities.

The milestone is achieved if the knowledge representation can be used to:

- o provide definitions of all terms, including how a term differs from closely related ones
- o explain distinctions and similarities between terms
- o state the purposes of a field, relation, file, or other data base structure
- o compare and contrast usage of terms in different data bases where similar terminology may arise

Since some work has already begun on the first two aspects above, and since the milestone does not seem to involve developing a new ontology (such as representing space and time symbolically does), a laboratory demonstration should be available in three years and a more tested demonstration in five years, using a small group of 2-3 person-years per year.

#### 4.4.3.2 Analogical reasoning

Reasoning by analogy is critical to making use of past experience. It includes not only discriminating between similar and dissimilar entities, but also being able to itemize (and explain) the factors in which two entities are analogous and the factors where they are <u>not</u> analogous.

The milestone is achieved if a system, given an archive of records regarding entities, can intelligently retrieve instances that are similar to an input description of an entity. Performance should be guided by how well an expert would:

- o supply only appropriate analogies
- o not miss relevant descriptions

- o adjust the number of analogous entities retrieved to user interest in accuracy and precision

- o explain or justify why an entity was analogous

This is probably best tackled in two steps.

1. Reasoning about analogous objects and events (when taken in isolation).

2. Reasoning about analogous plans and situations (when part of a sequence).

Since representation of knowledge about objects is more advanced than in the case of plans and situations, the first should be achievable in four years based on two very small groups or one modest-sized group with a total of 3-4 person-years per year. The second should be achievable in 7-10 years, since it depends on progress in research on the first step and on planning. Funding of 5 person-years per year starting four years from now is appropriate, so that two groups can investigate at least two different approaches. Analogy based on complex entities such as plans is much more demanding than for objects, which have less complexity.

There is an alternative school of thought that would argue that traditional information retrieval systems, i.e., keyword retrieval, is appropriate. We do not agree at all. The pilot/aircrew automation and $C^3$ scenarios all involve reasoning by analogy given an archive or data base of past tactics, plans, and situations. The systems design scenario assumes reasoning by analogy about typical mission plans, related design situations, and previous experience of personnel. Certainly, the systems design and $C^3$ scenarios also assume the ability to explain or justify analogies selected and rejected. Keyword retrieval seems an inadequate base for demonstrating the reasoning an expert might use in such situations.

## 4.5 COMPUTER VISION

### 4.5.1 Capabilities Required by the Scenarios

A. Recognizing features and objects from long-range sensors (PILOT/AIRCREW AUTOMATION, $C^3$). The key capability required here is recognizing entities sufficiently to indicate whether they are enemy, hostile or unknown. In addition to the obvious purpose of this, the scenarios make use of this to highlight the display of individuals monitoring the sensor data, so that they can focus their attention on items of interest, such as unidentified items or hostile items, and not have their display cluttered up with objects that are not of interest to them at the moment.

### 4.5.2 Current and Near-term State-of-the-art

The only capability in computer vision arising from the scenarios is in distinguishing hostile from friendly aircraft. Of course, it is difficult to obtain or report on capabilities for this particular capability, since the information is highly classified. Consequently, this discussion will only be in the broadest terms. The general state-of-the-art in computer vision is covered in section 2.5.

The DARPA Strategic Computing initiative has a different functional objective during the next five years: recognizing benchmarks and obstacles in a complex terrain. This could be applicable, for instance, as a navigational aid for cruise missiles.

A key simplification for both objectives is that man-made objects tend to be far easier to recognize than natural objects, due to regularity in form and structure, commonality of few (relative to natural objects) shapes, etc. Based on the level of commitment to the objective of the Strategic Computing initiative mentioned above, the degree of focus in the effort, and the simplifying aspect of recognizing man-made objects, we expect the probability of success in bringing technology to the point of being applicable at .8.

One of the biggest limitations to computer vision is not a vision problem at all. Of all the areas identified in this report, vision is the one most hampered by lack of processing power. Processing at least 1000 times faster is needed for computer vision.

Vision capabilities will be available from research labs, e.g., those involved in the Strategic Computing initiative, and from corporations, e.g., Machine Intelligence, and from universities with substantial vision laboratories.

#### 4.5.3 Milestones

##### 4.5.3.1 Aircraft recognition

To be able to distinguish hostile from friendly aircraft requires:

- o precise sensors, (e.g., radar and cameras)
- o a data base of sensor images of all aircraft types
- o adequate feature extraction and property identification strategies
- o processing speed for real-time recognition

The demands for real-time recognition and for accuracy are critical in this milestone but unfortunately are not common in AI research. No existing AI applications achieve them. As a consequence, we estimate 13 person-years per year as necessary to achieve this within the next five to ten years.

### 4.6 INTELLIGENT TUTORING & TRAINING SYSTEMS

#### 4.6.1 Capabilities Required by the Scenarios

AI work may be implied in the ISD part of the simulator design of the systems design scenario, but we have assumed it is not.

### 4.6.2 Current and Near-term State-of-the-art

Though no capabilities in the three scenarios directly depend on intelligent tutoring technology, it is clear, due to the amount and variety of training performed regularly for military needs, that advances in this field would be quite beneficial. Therefore, the state-of-the-art is reviewed in section 2.6 and milestones are provided here.

Few products are currently available. At one end of the spectrum is LOGO, a language designed to be used in education, especially noted for teaching geometry and programming concepts. Another commercially available set of programs are for authors and facilitate the writing of frame-based computer-assisted instruction (CAI) programs, which do not involve AI. Given the lack of promise of these frame-based techniques, authoring languages are of limited interest. However, their existence points out a gap in the tools available: a language or environment in which it would be possible for intelligent computer-assisted instruction (ICAI) software to be built easily by nonprogrammers.

Prototypes and demonstrations tend to involve so many components unique to the subject matter and educational goals that little building on old systems has occurred. Nor is there an emergence of standard architectures for systems. Oftentimes, research on an ICAI system necessitates that the project involve some research in knowledge representation, natural language processing, or expert systems. Consequently, the subject matter and style of ICAI goals is heavily dependent on advances in other areas of AI.

AI capability in this area for the next five years will come from research centers (e.g., Xerox PARC, BBN, Yale University). ICAI in general is not likely to succeed without breakthroughs in other areas of AI. Some limited applications of expert systems technology and natural language processing technology should be possible in the next five years. We believe the probability of this to be .8, due to the extensive growth in applied research in those areas and due to previous successes in transferring expert system technology and natural language technology to tutoring.

The gaps where research must be pushed are:

- o knowledge representation, because this plus expert systems and natural language technology provide the technological base upon which ICAI is based
- o expert systems
- o natural language processing
- o modeling student misconceptions, i.e., identification and definition of classes of errors, whether common or rare, is very labor intensive, but is critical for successful computer coaching
- o knowledge acquisition tools, such as in the case of expert systems, where software to bridge the gap between curriculum designer and AI programmer is a fundamental need
- o tools for generating example problems or exercises, e.g., generation of examples to be worked is an integral part of many types of problem-solving instruction
- o programming languages, tools, and techniques are important not only in the task of building the system but also in the style of student activity easily supported.

### 4.6.3 Milestones

#### 4.6.3.1 Multiple Strategies, One Domain

One milestone is a system which incorporates all of the basic techniques in an integrated fashion <u>in a single domain</u>. The key to the effort is integrating modeling, coaching, problem solving (in microworlds), inquiry teaching, and explanation into an effective environment. The architecture of this system will be a key research result, even though aspects of the design may be very subject specific.

This will probably occur in about 5-7 years, since systems exhibiting only one style typically require several person-years to develop. For this to occur, research on patterns or students' misconceptions in the chosen domain, on the value of alternative knowledge representations, and on coaching rules are all necessary conditions. A team of four is the minimum needed to build such a system; therefore, 4 person-years per year are necessary.

Some people may argue that drill-and-practice systems and frame-based systems are sufficient for educational purposes, since they provide immediate feedback and individualized instruction. This reflects an underlying difference of opinion about the nature of effective participation on the part of students. Our view is that AI offers substantially different kinds of tutoring, in particular, coaching of problem solving technique, planning, language use, etc.

## 4.7 PLANNING AND PROBLEM-SOLVING UNDER REAL WORLD CONDITIONS

### 4.7.1 Capabilities Required by the Scenarios

A. Typical attack/defense tactics *of an aircraft* (*PILOT/AIRCREW* AUTOMATION, SYSTEMS DESIGN, $C^3$). The problem here of course is having an adequate representation of the typical tactics of an aircraft. This means a rich enough model of plans, space, and time to represent those tactics.

B. Multiple goals (PILOT/AIRCREW AUTOMATION, SYSTEMS DESIGN, $C^3$). In real life situations, the desire to achieve many goals is not only common, but necessary. Current technology and planning finds achieving many goals difficult.

C. Typical sequences of bodily movements (SYSTEMS DESIGN). This capability involves representing stereotypical variations on performing tasks. This naturally presumes a fairly rich model of representing space as well as time.

D. Likelihood of success, given an adversary (PILOT/AIRCREW AUTOMATION, $C^3$). The problem in this capability is representing what the adversary might do, and representing how likely the adversary is to do any one of those countermeasures. Furthermore, it may or may not be adequate to view several sequences of move and counter move as a single tactic counter tactic pair. If the problem can be simplified in that way, it makes the capability much easier to achieve.

E. Coordination of diverse activities in real-time (PILOT/AIRCREW AUTOMATION, $C^3$). This capability involves knowing what activity needs most attention at the moment, what activities can be delayed for a short term, and what activities can be put in the background altogether. This is planning about what to think about and attend to.

### 4.7.2 Current and Near-term State-of-the-art

We are likely to have prototype planning in a military context as a result of the DARPA Strategic Computing initiative, within the next 5 years. We estimate the probability of success to be about .6 to .7, based on the difficulty of applying current technology with its restricting assumptions (e.g., single agent). These capabilities will be available from research labs (e.g., SRI International and BBN) and from recipients of the DARPA Strategic Computing contracts in expert systems.

Several planning capabilities are assumed by the scenarios; these are repeated from above:

o typical attack/defense tactics of an aircraft (PILOT/AIRCREW AUTOMATION, SYSTEMS DESIGN, $C^3$)

o multiple goals (PILOT/AIRCREW AUTOMATION, SYSTEMS DESIGN, $C^3$)

o typical sequences of bodily movements (SYSTEMS DESIGN).

o likelihood of success, given an adversary (PILOT, $C^3$).

o coordination of diverse activities in real-time (PILOT/AIRCREW AUTOMATION, $C^3$)

There are quite a number of underlying problems that must be solved in order to have those capabilities. It is these underlying problems that are the key to achieving the capabilities above. Factors which are particularly difficult are:

1. Appropriate representation of space and time, supporting both common sense reasoning and mathematical reasoning (see section 4.4 on Knowledge Representation). This arises in two of the capabilities above, viz., attack/defense tactics and sequences of bodily movements, since both involve actions in space and time.

2. Recognition that a plan is no longer valid or is now undesirable due to changing conditions. This results from the need to reason about attack/defense tactics. The environment will not remain static, due to natural events, enemy action, and friendly action; any changes could invalidate a plan.

3. Construction of revised plans as needed. Of course, if change of conditions invalidate an old plan, a revised plan is necessary regarding achieving a mission.

4. Explanation of why a plan has the form it does, rather than some other alternative. Explanation is necessary so that the commander, the pilot, etc. can know why an attack/defense plan was suggested. Otherwise, when the recommendation differs from the human's idea, there is no basis for comparing the two alternatives. (See the sections on Expert Systems and Natural Language Understanding).

5. Reasoning about how much computational resources to expend on a given problem. This is critical to the last capability above, coordinating diverse activities in real-time. That requires intelligent decisions regarding what to focus on and for how long.

6. Classification of individual problems as to whether special-purpose (e.g., algorithmic) methods or general-purpose search methods are more appropriate. Some of the capabilities above involve numerical reasoning. These include, for instance, trajectories, bodily movements, and numerical probabilities of success. Nevertheless, explaining decisions to an individual involves symbolic reasoning. Choosing the preferable technique for a given aspect of the capabilities is necessary.

7. Reasoning under uncertainty, including making plausible assumptions and retracting invalidated conclusions if the assumptions prove invalid. Such assumptions are necessary due to incomplete information. This arises not only in an adversarial environment but also in normal communication of requests via natural language. (See also Section 4.4 on Knowledge Representation).

Items one, two, three, five, and seven are covered in the first milestone below. Item four is part of the milestones on natural language. Item six is implicit in the second milestone below.

### 4.7.3 Milestones

#### 4.7.3.1 Common Sense System

By this we mean a system with:

- appropriate representation of space and time, supporting both common sense reasoning and mathematical reasoning
- recognition when a plan is no longer valid or is now undesirable due to changing conditions
- construction of revised plans as needed
- reasoning about how many computational resources to expend on a given problem
- reasoning under uncertainty, including nonmonotonic reasoning, i.e., making plausible assumptions and retracting invalidated conclusions if the assumptions prove invalid
- planning in the context of multiple goals

These problems are critical to military applications, in particular to the pilot/aircrew automation scenario and to the $C^3$ scenario. Furthermore, they are sufficiently difficult that the research should be based on far simpler domains or environments than envisioned in either the pilot/aircrew automation or $C^3$ scenarios. Rather, a simplified domain, perhaps patterned after the systems design scenario is far more appropriate.

Particularly in the initial years, the effort can be partitioned into independent efforts, such as:

- representing space and time plus encoding common sense knowledge as axioms
- plan revision
- reasoning about devoting resources to planning
- nonmonotonic reasoning

This milestone involves several fundamental breakthroughs (e.g., nonmonotonic reasoning, explicit control of computational resources, revising plans, etc.), which are not incremental extensions of existing technology. Therefore, we expect a demonstration of this milestone to be achievable in 8 years at the earliest. More applicable versions should be available within 5 years after that. Funding is clearly inadequate for this, since at most two of the factors above are being addressed currently, and since common sense reasoning has been one of the hardest, most central problems in AI. Consequently, we recommend additional effort at the level of 8 person-years per year; this would allow two or three teams to work on various aspects using different hypotheses. Examining competing hypotheses is particularly important here, since these are not extensions to existing technology.

There are no alternative points of view regarding the goals of this milestone, though there is debate about whether employing numerical certainty factors is necessary or even advisable. It is important to recognize that there is little consensus on frameworks, by contrast with the situation in such better understood domains as syntax in natural language processing. Rather, research in the past has often been led by the structure of the particular application domain being studied. This is a reason to emphasize that the research minimize domain dependence.

As a consequence, a major goal of the funding should be that the principles in building the reasoning system should be independent of peculiarities of the domain and should be applicable independent of domain. The detailed knowledge may be domain-specific.

#### 4.7.3.2 Planning against Adversaries

This milestone represents research that could be used by an aircraft pilot to aid in mission planning as in the pilot/aircrew automation scenario or by a commander using a $C^3$ aid to project possible adversarial action. The system would know enough about airplane characteristics, a pilot's characteristics, mission goals, and environmental characteristics, so that it could monitor all of these components for changes that might affect current plan execution, and provide for replanning to achieve the mission in case the initial plan is invalidated. It should use special-purpose methods for certain subproblems, e.g., computing trajectories, where appropriate. These don't necessarily give a better solution, but they may give a solution more efficiently, which is essential in a combat situation.

Since planning assumes the common sense reasoner in the previous milestone and since this requires real-time performance of a very high caliber, we expect it will be 10-12 years before the technology will exist to produce the first demonstrable systems. Additionally, to have a rich model of adversarial plans, and to project adversarial action in both the $C^3$ and pilot/aircrew automation scenarios is probably 15-18 years away, since this introduces an additional level of complexity not investigated before.

The major alternate school of thought is one that doesn't try to use "heuristic" or general-purpose planning procedures at all, but tries to translate all planning problems into some framework on which special-purpose methods can be used, e.g., methods of numerical analysis. This seems too extreme a point of view to us.

To achieve the milestone by the date specified, it will probably be necessary to increase the level of effort substantially, particularly to explore solutions to the problems mentioned in the section on current capabilities. We suggest:

- an additional 7 person-years per year while work on the common sense milestone progresses to support two modest-sized teams
- 13 person-years per year after achieving the first milestone, so that two somewhat large teams can work on the problem

## 4.8 AI TOOLS AND ENVIRONMENTS

### 4.8.1 Capabilities Required by the Scenarios

Though the capabilities do not require any particular demands on AI tools and environments, AI tools and environments are at the heart of all of the capabilities, for this is the infrastructure upon which all AI research, AI development, and AI applications are built.

### 4.8.2 Current and Near-term State-of-the-art

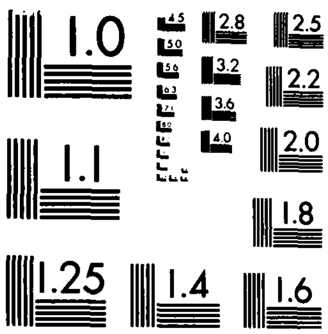
No capabilities in this area are specifically called out by the scenarios. However, since this area is the infrastructure upon which both AI applications and research are based, it is unquestionably important. Processing speed is particularly critical for real-time vision, planning, speech recognition, and expert systems. Programming environments that improve the productivity of individuals are critical in all of the areas, since all of the efforts (both research and application) are labor intensive.

The main systems currently available are Lisp-based and will continue to be extended both in functionality and performance. Costs will also clearly fall due to VLSI advances in the next few years. Programming environments will become more robust for the other paradigms, such as Prolog. These developments will mostly occur in companies such as Xerox, Symbolics, and BBN. Probability of success here is .9, since these are incremental improvements using existing technology.

Basic research must continue in the exploration of programming paradigms and their integration and unification. This being more speculative, it will occur largely in universities or research institutions. Also, the probability of success decreases with the degree of speculative nature, with high probability, say .8, for current paradigms, since this is evolutionary progress. The probability is perhaps .5 or less in new programming paradigms,

AD-A161 139 AUTOMATED INFORMATION MANAGEMENT TECHNOLOGY (AIM-TECH): 3/3
CONSIDERATIONS FO (U) MACAULAY-BROWN INC FAIRBORN OH
W REITMAN ET AL MAY 85 AFAMRL-TR-85-042
UNCLASSIFIED F33615-82-C-0513 F/G 9/2 NL

END
FILMED
DTIC




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

and the integration and/or unification of several paradigms, since this involves not only technological advances but also adoption of the new paradigms in a large portion of the community's programming practices.

### 4.8.3 Milestones

#### 4.8.3.1 Conventional Work stations

A Lisp machine an order of magnitude faster than today's should be available for $35,000 by 1986 or 1987. This assumes technological development in hardware based on directing efforts towards VLSI versions of a Lisp machine. By comparison, Lisp machines with today's capabilities should be available then for $10,000 or less. This milestone should be accomplished given current levels of funding, particularly industrial research support.

#### 4.8.3.2 Highly parallel programming

A robust programming environment on some highly parallel machine should be available by 1987-1988, via introducing limited parallelism into existing programming languages. An example is Multi-Lisp being developed at BBN for the BBN Butterfly multiprocessor. However, it will probably not be until 1995 or 2000 before both highly parallel hardware and software are understood and exploited as well as sequential programming is now. This presumes conceptual breakthroughs on how to express algorithms and heuristics to take advantage of parallelism in an effective way. It does not appear that such breakthroughs are near; furthermore, highly parallel machines in the past have been used effectively on highly specialized problems only.

There are several schools of thought regarding parallelism. One is to design a machine based on unconventional architectures that have arisen in AI software. Example proposals are production rule architectures (Allen Newell at CMU) and marker-passing and activation networks (Scott Fahlman at CMU, Marvin Minsky at MIT, Daniel Hillis at Thinking Machines Corporation). The question with such an approach is how many AI applications can be conveniently modeled with such an architecture, where "convenient" means taking advantage of the architecture rather than building a von Neuman machine on top of it.

A second view centers on programming language research and views the problem as one of expressing the parallelism of an algorithm or heuristic. Much work on applicative programming (though not usually oriented to AI applications) and that of Carl Hewlett (MIT) and N. Sridharan (BBN) falls into this category.

A third view is to evolve hardware and software from well-known sequential structures. The advantage of this is that there is no radical break with current practices; instead, usable products should be available at each stage. D. Allen of BBN is taking this approach.

The fourth and fifth views are closely related in that they are conservative with respect to parallelism. In the fourth view, AI problems that are at least moderately well understood are reexamined for parallelism; examples include natural language understanding (R. Bobrow at BBN) and parsing (J. Robinson at SRI International). Given that this is more conservative in approach, chances of success are far higher.

The fifth view agrees that, while research according to the other three views should proceed, it is not fundamental to solving AI problems. Rather, according to this argument, what is fundamental are the AI problems. Many involve exploring numbers of alternatives that grow radically as the size of the input increases. What is needed, they argue, are intelligent ways to focus on few alternatives the way an expert would.

Many alternative architectures for parallel computation have been proposed, ranging over number of processors, complexity of processors, bandwidth of coupling processors, interaction of processors, etc. The methods of controlling these various architectures will likely have little to do with each other.

In one sense, no amount of effort could be too much, since the problems are so complex and there has been so little progress. On the other hand, substantial levels of support are available from other sources. There is a greater need for funding of the fourth and fifth views listed above, than for the first, second, and third, given current funding patterns. The milestones in natural language and in planning provide implicitly some additional funding for the fourth and fifth views. An additional 2 person-years per year for each of the fourth and fifth views would allow an additional very small team to work on each.

## 4.9 SPEECH

### 4.9.1 Capabilities Required by the Scenarios

A. Understanding and synthesis over a broad domain (SYSTEMS DESIGN, $C^3$). This capability involves not only the problems of speech, but also the problems of natural language processing. Because a broad domain is involved, all the problems of attaining natural language in a broad domain are true here as well.

B. Understanding and synthesis in a limited domain (PILOT/AIRCREW AUTOMATION). The advantages of dealing with a limited domain are that not all complications of full natural language may arise. In fact, it is possible in the limited domain of a cockpit that a very limited range of syntactic patterns and a very limited vocabulary will suffice. Furthermore, it is possible that the number of tasks to be carried out as a result of a speech command will be relatively small.

### 4.9.2 Current and Near-term State-of-the-art

Based on the expectation of cheaper hardware, special purpose hardware, and incremental improvements in algorithms, we can confidently project that with probability .8 the trends shown by current speech recognition systems will continue. This assumes that the current base of funds from commercial sources and the government, such as DARPA's Strategic Computing initiative continues. Specifically, individual word recognition (IWR) systems will continue to become cheaper, handle larger vocabularies, achieve higher performance, and progress toward speaker independent recognition, at least for smaller vocabulary sizes. Connected word recognition (CWR) systems will progress similarly, and grammatical constraints of tasks will become more widely applied. These trends will be driven primarily by cheaper computation and by incremental research driven by this availability of computation. Therefore, those applications that can be served by IWR and CWR systems will be served more effectively.

However, the limitations of the word-based approaches described earlier will become felt as these systems attempt to grow toward high-complexity applications. Speech understanding is not simply a matter of stringing words together, and higher level linguistic knowledge will be required both to recognize and to utilize fluently spoken interactions in advanced applications. We can expect current research in phonetically based speech recognition to support the lower levels of speech understanding, but the broader support of other knowledge sources will need to be revived for advanced speech understanding to become possible. Particularly, advances in natural language processing, in integration of constraints in natural language settings, and in low-level phonetic recognition (to enable speaker independence) are needed. These needed capabilities are not available from computer science alone, but must also include signal processing, speech communications, linguistics, and cognitive science.

The capabilities assumed by the scenarios are for continuous speech recognition and synthesis in a limited domain and also in broad domains. We have split these into three milestones.

### 4.9.3 Milestones

DARPA's Strategic Computing initiative includes goals in speech recognition, and this program is now in procurement. These goals depict two types of systems to be developed within 10 years: small vocabulary CWR and large vocabulary CSR.

#### 4.9.3.1 Small vocabulary, CWR

The first milestone is a 200-word CWR system, capable of running in real-time in an environment with a restricted grammar, speaker independence, severe noise, severe psychological and/or physical stress.

Such a speech recognition system is intended for an application of assisting an aircraft pilot. The vocabulary is small and the task-oriented commands have a rather constrained syntax. The speech will be uttered in a very noisy environment, and the speaker will be subject to G-forces, and emotional stress. Such a system must be small enough to be installed in aircraft. However, it would also have practical application in very limited tasks of battle management aids and limited natural language query of data bases.

This milestone should be achievable in 7-8 years given projected support. This assumes the availability of suitable fast computation or special purpose VLSI designs, research on noise handling (signal processing, recognition); and research on speaker stress (speech production, recognition), in addition to the effort of assembling the results of the parallel research efforts into a system.

Additionally, this assumes that such a system can be achieved using present grammar-driven CWR techniques with multiple clustered templates per word; that the signal analysis and recognition can be extended to handle the noise, and that the effects of stress can be characterized and also handled by multiple templates. Other points of view would probably stress the recognition of smaller units such as phones and characterizing the effects of stress and speaker differences on them.

There is much industrial support for this kind of capability, as well as the Strategic Computing initiative. This will probably be adequate.

#### 4.9.3.2 Large vocabulary continuous speech recognition (CSR)

The second milestone is a 10,000-word CSR system with a limited natural language grammar, some speaker independence, moderate noise, low stress, and multiple knowledge sources (e.g., grammar, limited pragmatics, etc.).

This kind of speech recognition system is intended for person-machine interaction in a much more habitable environment, such as situation assessment and management. The vocabulary and grammar approach limited natural language, and therefore provide much less constraint than in the Pilot's Associate.

This milestone's feasibility should be demonstrated in 4-5 years. It should be achievable in 10 years. This projection assumes research advances in many areas, such as phonetic/subword recognition, prosodics, syntax, semantics, pragmatics, speaker difference, and use of these multiple, diverse knowledge sources. In addition, extensive speech data bases, and appropriate hardware arising from advances in VLSI technology and multiple architectures are necessary.

Because of the multiplicity of problems to be solved, increased support would be useful, but if effects are to be seen in the near future, research and development must be carefully managed and even coordinated. The more labs working on this the greater the chances of success. Each lab should be funded at 4-7 person-years per year, so that a modestly large team can work on the diverse areas listed above.

Real-time operation is not foreseen until several years after this milestone.

#### 4.9.3.3 Continuous speech understanding in a broad domain

The kind of capability foreseen in the systems design scenario and $C^3$ scenario is probably 20 years away. This is predominantly due to the need for substantial breakthroughs in natural language processing to achieve coverage in a broad domain.

Funding is needed initially for progress in natural language processing; see section 4.3 on Natural Language. Additionally, after progress is made in pragmatics, particularly in understanding speaker intentions and in processing ill-formed and errorful input (since speech contains a high frequency of ungrammaticality and since low level speech processing algorithms will make errors), funding for integrating the natural language technology into speech systems is needed.

Assume that the other milestones in speech and natural language are being funded as suggested. Then, an additional one or two labs funded specifically on the long term problems of integration is necessary. We recommend 7 person-years per year for the first 5 years, and 13 person-years per year thereafter. The initially low figure is to support a modestly large team concentrating on the speech-specific problems, while natural language research progresses separately. After 5 years, progress in natural language should make research in the interface between the two technologies possible. The higher funding level could support two modestly large teams to investigate two differing research paths.

## Section 5. CONSIDERATIONS FOR A TECHNOLOGY INVESTMENT STRATEGY

Kenneth R. Boff, PhD
Ralph M. Weischedel, PhD

### 5.1 INTRODUCTION

Our approach to defining a technology investment strategy has been to first identify those AI areas which might contribute to the solutions of Air Force problems if they were better understood. These are documented in Section 2.0 of this report. Air Force problems were defined within the context of technical domains: systems design, pilot/aircrew automation, and $C^3$ (detailed in Section 3.0). Futuristic scenarios were constructed in which information and control management choke points are resolved by hypothetical application of machine intelligence in each of these domains. The technology demands of each scenario were then analyzed and matched with technology milestones identified in each AI area, and estimates of when the milestones might be achieved (detailed in Section 4.0). Hence, the investment strategy consists of a system for comparing relevant investment opportunities against the level of effort needed to achieve a technology breakthrough in each. The main objective of this section of the report is to help assure that future funding by AFAMRL of research and development of AI technology will provide return on investment in terms of benefits of this technology to aiding the solution of Air Force problems encompassed by the three scenarios. Data are provided to support selection of target areas for funding by providing a basis for quantifying investment decisions, thereby allowing potential value to be estimated against cost.

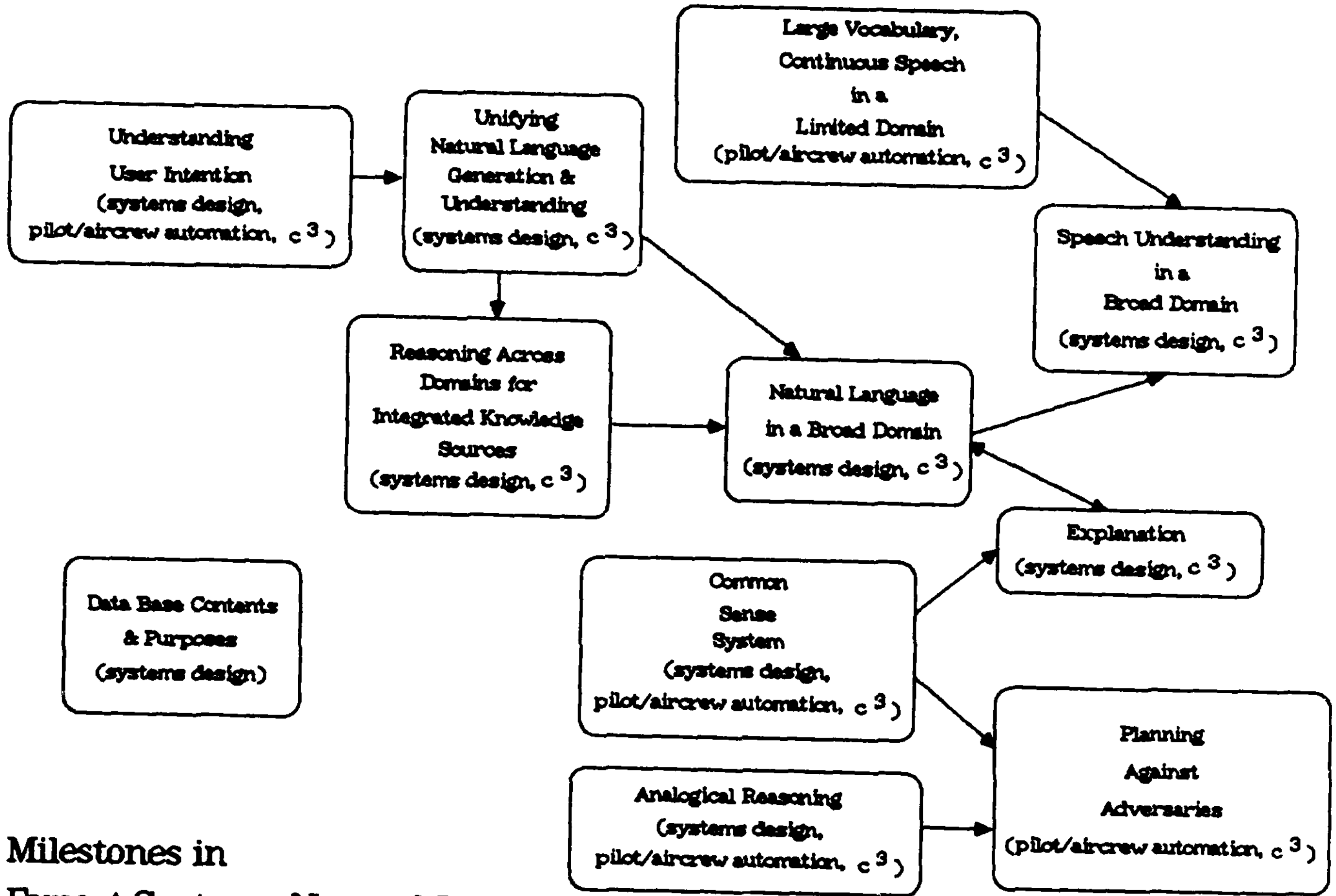
### 5.2 TECHNOLOGY DEMANDS OF THE SCENARIOS

Though the report divided the capabilities evident in the three scenarios into eight subareas of AI research, few of the capabilities could be achieved by progress in a single area. Rather the capabilities assume progress in highly interrelated areas, as evidenced by the number of cross-references in the text.

As an example of this, consider the expert systems envisioned in the scenarios. Though current expert systems research has not been closely tied to other areas, it is clear that the expert systems envisioned in the scenario do. They have sophisticated capability both in natural language understanding (to provide real-time, direct control by the user, e.g., designer, pilot, or commander) and in natural language generation (to justify why a recommendation is made and to communicate both rapidly and effectively to the user). Those expert systems provide planning capabilities to achieve the mission and user goals. They depend on knowledge representation to support reasoning with incomplete information, to support advice regarding relevant data bases, and to enable reasoning by analogy from encoded, past experiences.

Consequently, fundamental advances in many subareas are needed to support the scenarios. Figures 3 and 4 and Table 3 summarize the milestones needing additional support and effort. Three milestones described in Section 4.0 are not included in the summary, since our projections are that they will be achieved given currently projected support, such as industrial research and the DARPA Strategic Computing initiative. These are a complete natural language system in a well-scoped, practical domain (discussed in Section 4.3); advances in conventional AI work stations (discussed in Section 4.8); and small vocabulary, connected word recognition systems (discussed in Section 4.9). All three are either directly assumed by each of the scenarios or indirectly assumed in that they contribute to achieving other milestones that are assumed. The figures and table include only one aspect of the milestone of integrating knowledge sources (discussed in Section 4.2). Only the aspect of reasoning across domains is included, since the other two aspects, integrating expertise from various mathematical bases and integrating reasoning from differing knowledge sources in one domain, should be achieved with currently projected effort. These two aspects are necessary to the pilot/aircrew automation and systems design scenarios.

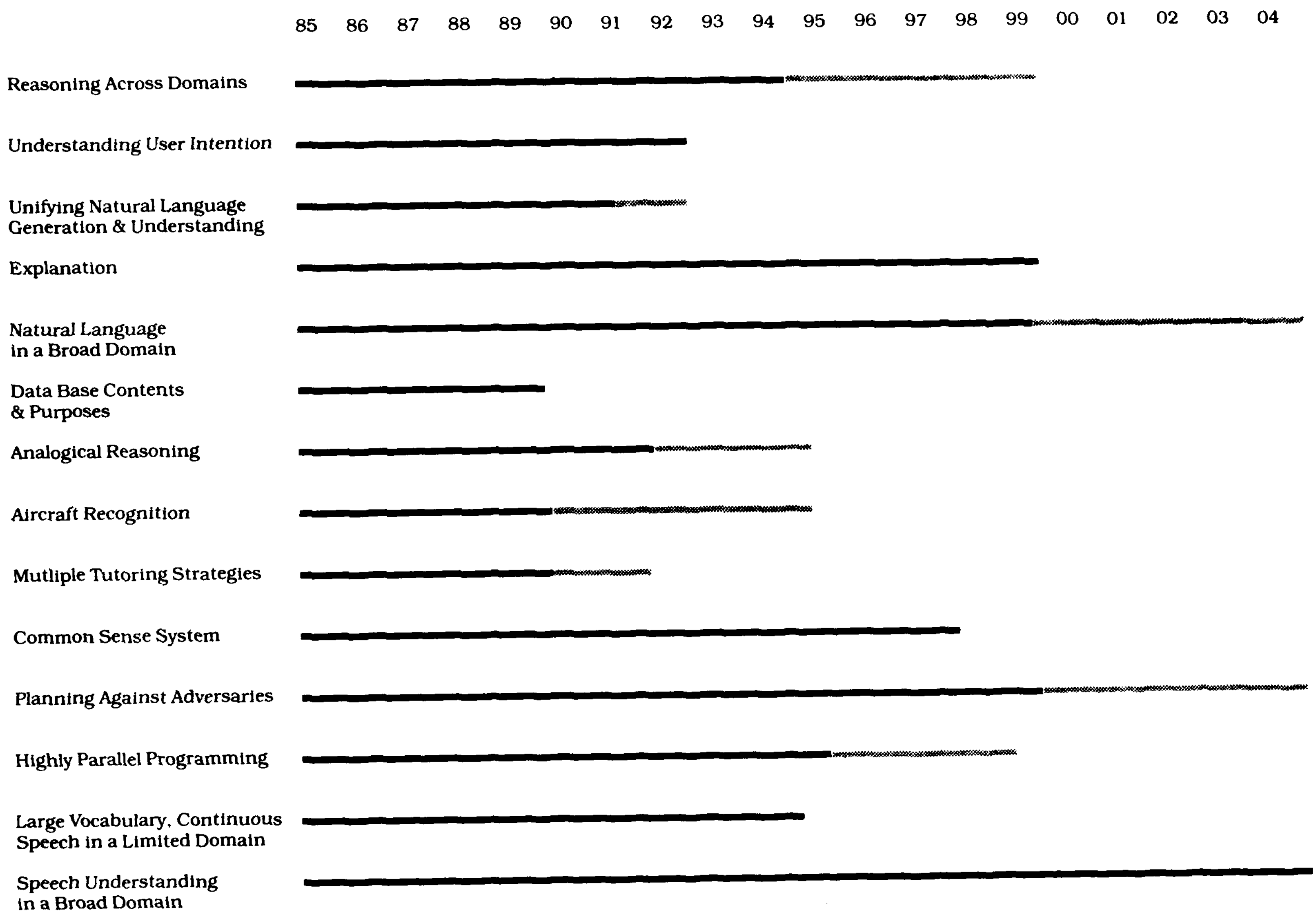
A summary of the logical dependencies among the milestones in expert systems, natural language, knowledge representation, planning, and speech




Milestones in
Expert Systems, Natural Language,
Knowledge Representation, Planning, and Speech

Figure 3

(Arrows indicate logical dependence, and therefore that progress in one milestone is necessary for the completion of the one at the head of the arrow.)




Time Line for Milestones

Figure 4

TABLE 3. COST AND LEVEL OF EFFORT ESTIMATES FOR THE PROJECTED TECHNOLOGY OBJECTIVES

| Milestone/Technology Demand | AI Area | Annual Estimated Person-Years Per Year (Inclusive) For Level Of Effort*1 | Total Estimated Level Of Effort*1 (Person-Years) | Total Estimated Cost*2 (Millions Of Dollars) |
|---|---|---|---|---|
| Full-Scale Explanatory Capability | Expert Systems | 2 ('85-'86)<br>5 ('87-'92)<br>10 ('93-'99) | 99 | 14.850 |
| Reasoning Across Domains for Integrated Knowledge Sources | Expert Systems | 3 ('85-'89)<br>6-7 ('90-'99) | 75-85 | 11.250-12.750 |
| Planning in a Rapidly Changing Environment with Adversaries | Expert Systems | 7 ('85-'96)<br>13 ('97-'02) | 162 | 24.300 |
| Understanding User Intention | Natural Language Processing | 5 ('85-'92) | 40 | 6.000 |
| Unifying Natural Language Generation and Understanding | Natural Language Processing | 3 ('85-'86)<br>6-7 ('87-'92) | 42-48 | 6.300-7.200 |
| Explaining and Paraphrasing | Natural Language Processing | 2 ('85-'86)<br>5 ('87-'92)<br>10 ('93-'99) | 99 | 14.850 |
| Natural Language in a Broad Domain or Across Narrowly-Scoped Domains | Natural Language Processing | 3 ('85-'92)<br>20 ('93-'04) | 364 | 54.600 |
| Data Base Content and Purposes | Knowledge Representation | 2-3 ('85-'89) | 10-15 | 1.500-2.250 |
| Analogical Reasoning | Knowledge Representation | 3-4 ('85-'88)<br>5 ('89-'94) | 42-46 | 6.300-6.900 |
| Aircraft Recognition | Computer Vision | 13 ('85-'94) | 130 | 19.500 |
| Multiple Tutoring Strategies | Intelligent Tutoring and Training Systems | 4 ('85-'91) | 28 | 4.200 |
| Common Sense System | Planning | 8 ('85-'97) | 104 | 15.600 |
| Planning Against Adversaries | Planning | 7 ('85-'96)<br>13 ('97-'02) | 162 | 24.300 |
| Highly Parallel Programing | AI Tools and Environments | 4 ('85-'00) | 64 | 9.600 |
| Large Vocabulary, Continuous Speech in a Limited Domain | Speech | 4-7 ('85-'94) | 40-70 | 6.000-10.500 |
| Speech Understanding in a Broad Domain | Speech | 7 ('85-'89)<br>13 ('90-'04) | 230 | 34.500 |

*1 These estimates are for additional effort over the current funded baseline.

*2 Cost is figured on $150,000 Per Person-Year in FY84 dollars.

is provided in Figure 3. Arrows indicate the directions in which results must flow.

The time frame in which the various milestones will be achieved is summarized in Figure 4, and the annual additional support needed to achieve those milestones is summarized in Table 3. Estimates of annual additional person-years needed are also provided. Given typical salaries, work station costs, etc., it is not unreasonable to project that a person-year in AI research now requires roughly $150,000 on the average in terms of 1984 dollars. A range of $125,000 to $175,000 is realistic and reasonable.

What has been provided is analysis and estimates of research needed to provide capabilities. The scenarios will be achieved only after development effort and field testing once research has made the capabilities achievable. Development efforts and field testing are not analyzed here.

## 5.3 CONSIDERATIONS FOR AN INVESTMENT STRATEGY

Each of the scenarios (Section 3.0) emphasized technology endpoints that in turn pose technology demands which probably exceed realistic near term requirements in each of these domains. Hence, the high projected cost and level of effort needed to meet these technical demands should not be surprising. These data are summarized in Tables 4-6 which show the resource requirements needed to meet the technology demands of each scenario. These tables consider each AI area in terms of: estimated probability of success in meeting the technical demands of the scenario; relevant milestones/technical demands specified for each scenario; BBN projected levels of effort needed to achieve the required technology breakthrough; and the time period over which this level of effort must be distributed. The actual weighting for this distribution over time is shown in Table 3. Also, the projected level of effort is that needed over and above the current FY85 funded baseline.

TABLE 4. REQUIREMENTS FOR ACHIEVING THE SYSTEMS DESIGN SCENARIO

| AI Area | Probability of Success | Milestone/Technology Demand | Total Estimated Level of Effort (Person-Years)* | Period (Years) |
|---|---|---|---|---|
| Expert Systems | .9 | A Full-Scale Explanatory Capability | 99 | 15 |
| | | Reasoning Across Domains for Integrated Knowledge Sources | 75-85 | 10-15 |
| Natural Language Processing | .8-.9 | A Well-Scoped, Practical Domain | -- | -- |
| | | Understanding User Intention | 40 | 4-8 |
| | | Unifying Natural Language Generation and Understanding | 42-48 | 6-8 |
| | | Explaining and Paraphrasing | 99 | 17 |
| | | Natural Language in a Broad Domain or Across Narrowly-Scoped Domains | 364 | 15-20 |
| Knowledge Representation | .8-,9 | Data Base Content and Purposes | 10-15 | 3-5 |
| | | Analogical Reasoning | 42-46 | 4-10 |
| Planning | .6-.7 | Common Sense System | 104 | 8-13 |
| AI Tools and Environment | .5-.9 | Highly Parallel Programming | 64 | 10-15 |
| Speech | .8 | Continuous Speech Understanding in a Broad Domain | 230 | 20 |

* These estimates are for additional effort over the current funded baseline.

TABLE 5. REQUIREMENTS FOR ACHIEVING THE PILOT/AIRCREW AUTOMATION SCENARIO

| AI Area | Probability of Sucess | Milestone/Technology Demand | Total Estimated Level of Effort (Person-Years)* | Period (years) |
|---|---|---|---|---|
| Expert Systems | .9 | A Full-Scale Explanatory Capability | 99 | 15 |
| | | Reasoning Across Domains for Integrated Knowledge Sources | 75-85 | 10-15 |
| | | Planning in a Rapidly Changing Environment with Adversaries | 162 | 10-18 |
| Natural Language Processing | .8-.9 | A Well-Scoped, Practical Domain | -- | -- |
| | | Understanding User Intention | 40 | 4-8 |
| Knowledge Representation | .8-.9 | Analogical Reasoning | 42-46 | 4-10 |
| Computer Vision | .8 | Aircraft Recognition | 130 | 5-10 |
| Planning | .6-.7 | Common Sense System | 104 | 8-13 |
| | | Planning Against Adversaries | 162 | 10-18 |
| AI Tools and Environment | .5-.9 | Highly Parallel Programming | 64 | 10-15 |
| Speech | .8 | Large Vocabulary Continuous Speech Recognition in a Limited Domain | 40-70 | 4-10 |

* These estimates are for additional effort over the current funded baseline.

TABLE 6. REQUIREMENTS FOR ACHIEVING THE COMMAND, COMTROL, AND COMMUNICATION SCENARIO

| AI Area | Probability of Success | Milestone/Technology Demand | Total Estimated Level of Effort (Person-Years)* | Period (Years) |
|---|---|---|---|---|
| Expert Systems | .9 | A Full-Scale Explanatory Capability | 99 | 15 |
| | | Reasoning Across Domains for Integrated Knowledge Sources | 75-85 | 10-15 |
| | | Planning in a Rapidly Changing Environment with Adversaries | 162 | 10-18 |
| Natural Language Processing | .8-.9 | A Well-Scoped, Practical Domain | -- | -- |
| | | Understanding User Intention | 40 | 4-8 |
| | | Unifying Natural Language Generation and Understanding | 42-48 | 6-8 |
| | | Explaining and Paraphrasing | 99 | 17 |
| | | Natural Language in a Broad Domain or Across Narrowly-Scoped Domains | 364 | 15-20 |
| Knowledge Representation | .8-.9 | Analogical Reasoning | 42-46 | 4-10 |
| Computer Vision | .8 | Aircraft Recognition | 130 | 4-10 |
| Planning | .6-.7 | Common Sense System | 104 | 8-13 |
| | | Planning Against Adversaries | 162 | 10-18 |
| AI Tools and Environment | .5-.9 | Highly Parallel Programming | 64 | 10-15 |
| Speech | .8 | Large Vocabulary Continuous Speech Recognition in a Limited Domain | 40-70 | 4-10 |
| | | Continuous Speech Understanding in a Broad Domain | | |

* These estimates are for additional effort over the current funded baseline.

These data are also instructive in other ways.

- o They provide an almost shocking infusion of reality regarding the resources it will take to develop machine intelligence to the integral system component level emphasized by the scenarios. By providing a scaled endpoint along which AI technology has been projected, these data may provide rough guidance for the costs of achieving interim realities or scenarios in which less demand is made on the role of machine intelligence in the system. Developing an interim goal will require careful scoping and analysis of the characteristics/technology demands of the shared intelligence system. Sections 2.0 and 4.0 of this report can then be used for determining a reference baseline for projected technology goals. It is also recommended that the R&D manager attempting to project a new baseline consult with Martino (1983) for additional guidance. Hence, these data provide the prospective technology manager a limited basis for developing a return on investment analysis.
- o Table 7 shows that there is a considerable degree of overlap in the milestones/technology demands between scenarios. To some extent, this is an artifact of the resolution limits of the analysis and the specificity of the scenarios. Further analysis is needed for each of these milestones with respect to the scenarios in order to assess its relative weight or criticality to achieving the scenario. Nonetheless, investment in areas overlapping across the three scenarios are likely to show return on investment in more than a single domain.

### 5.3.1 Caveats For Interpreting These Data

- o Unspecified technology demands. Because of the resolution limits in analyzing the scenarios as well as the specificity of the scenarios themselves, it is probable that many implied technology demands were not addressed by our analysis. Also the scenarios involve considerable interactions among the AI technologies. Many of the residual issues associated with achieving such an

TABLE 7. MILESTONES/TECHNOLOGY DEMANDS AND THEIR RESPECTIVE SCENARIOS

| Milestones/Technology Demands | Scenarios Affected | | |
|---|---|---|---|
| | Systems Design | Pilot/Aircrew Automation | Command, Control and Communications |
| Full-Scale Explanatory Capability | X | X | X |
| Reasoning Across Domains for Integrated Knowledge Sources | X | | X |
| Planning in a Rapidly Changing Environment with Adversaries | | X | X |
| Understanding User Intention | X | X | X |
| Unifying Natural Language Generation & Understanding | X | | X |
| Explanation | X | | X |
| Natural Language in a Broad Domain | X | | X |
| Data Base Content & Purposes | X | | |
| Analogical Reasoning | X | X | X |
| Aircraft Recognition | | X | X |
| Multiple Tutoring Strategies | (no analysis given, See Section 4.6.2) | | |
| Common Sense System | X | X | X |
| Planning Against Adversaries | | X | X |
| Highly Parallel Programming | (infrastructure, See Section 4.8.2) | | |
| Large Vocabulary, Continuous Speech in a Limited Domain | | X | X |
| Speech Understanding in a Broad Domain | X | | X |

integration are not well understood at present and have not been dealt with in this report. These will require more detailed consideration before defining an investment strategy for an interim system capability. These may be classified into demands on machine intelligence and other technologies (e.g., sensor capabilities, maturity of knowledge bases such as "training effectiveness", etc.)

- Availability of AI professionals. A major concern arising from this analysis is the availability of sufficient competent personnel to meet the projected level of effort demand. This is likely to be an escalating problem as a broad range of developing demands compete for scarce personnel resources. This problem will be further exacerbated by the growing number of academics leaving full-time positions at training institutions for more lucrative industrial sector careers.
- Current maturity of AI technology. The exponential growth and immaturity of the current state of artificial intelligence as a technical discipline provides a rather unstable basis from which to make "accurate" predictions. An unprojected breakthrough that has commercial value could rapidly snowball the current levels of investment and intensify the current R&D baseline in an unpredictable fashion. Therefore, the half-life of the projection made in this report will vary with the occurrence of major AI technology events in the future.

### 5.3.2 Style of Projects

Some comments regarding recommended style of research projects are important as well. There is already substantial funding for constructing lab demonstrations and also for very small efforts that address a single issue. In our view, the overwhelming bulk of funds should continue to be devoted to these two styles of research. However, a third kind of R&D project is intermediate in size and involves focus on a particular task for a short period of time. An example might be getting a half dozen scientists together for a summer or half a year to develop a body of axioms about common sense reasoning about space; the payoff would be knowledge base which is potentially usable not only in each of the three scenarios but also in many other axioms. Such a knowledge base should be much more easily adapted to a new application than program code, since it is specified with less programming detail.

Since many of the problems that must be addressed are fundamental, long term, generally application-independent, and although their solution is essential to military applications in addition to the scenarios, it would be a mistake to focus R&D resource on building complete systems, e.g., the designer work stations, until substantial progress is made on the individual problems. Our recommended milestones reflect this; individual contracts certainly need not address all aspects of a milestone until near the projected accomplishment of the milestone.

Another style of research that is currently not adequately funded is coordinated, cooperative projects at more than a single institution. Usually, experts on a given topic, such as anaphora or representation of time, are spread over a number of institutions. When such resources can be pooled, research will advance more rapidly.

One related aspect is development of components that can be shared in the research community. By this we do <u>not</u> mean the adoption of a standard programming language. Instead, software components that are labor-intensive could be shared to substantially reduce research costs in some selected areas. DARPA has supported some efforts along these lines, e.g., the RUS parser of English from BBN and the KL-TWO knowledge representation language, both from BBN and USC Information Sciences Institute. Of course, careful selection of projects to support along those lines is critical; it does little good to support a component which no one wishes to use. Consequently, one must target tools that are likely to be usable elsewhere (i.e., because there is agreement among a potential community of users on much detail about the component) and that appear near a mature plateau. The component need not have an eternal life cycle; in fact, after a few years of use, new technology may suggest a new approach. We feel funding a few areas over a period of years is necessary for this purpose; current examples include:

- a declarative grammar of dialogue and associated lexicon, useful for both understanding and generation
- a related, declarative grammar of written prose and associated lexicon, useful for both understanding and generation
- a collection of axioms of common sense knowledge about space, time, physical objects, etc.
- a catalog of dialogue plans that are typically used in interactive communications

### 5.3.3 Recommended Style of Funding

The problems that most need work are of such a fundamental nature that the most promising minds (independent, of course, of seniority) are needed. The number of such individuals is not likely to grow as fast as the research funding, particularly given the rapid rise in industrial support. Consequently, a model we suggest is increasing the amount of support per individual, so that the computational support, number of programming staff, secretarial support, etc., enable the scientist to work at maximum potential. This includes providing the most up-to-date hardware/software on a continuing basis as hardware advances become available.



U S GOVERNMENT PRINTING OFFICE 1986 - 646-006/20219

END

FILMED

12-85

DTIC