NASA Contractor Report 3944

An Analysis of the Application of AI to the Development of Intelligent Aids for Flight Crew Tasks

Sheldon Baron and Carl Feehrer

CONTRACT NAS1-17335
OCTOBER 1985

NASA

NASA Contractor Report 3944

An Analysis of the Application of AI to the Development of Intelligent Aids for Flight Crew Tasks

Sheldon Baron and Carl Feehrer
Bolt Beranek and Newman Inc.
Cambridge, Massachusetts

Prepared for
Langley Research Center
under Contract NAS1-17335

NASA
National Aeronautics
and Space Administration

Scientific and Technical Information Branch

1985

TABLE OF CONTENTS

1. INTRODUCTION

Modern microprocessor and display technologies have made it feasible to automate many flight deck functions. The recent advances in artificial intelligence (AI) suggest that it may have a great deal to offer in the way of providing advanced automation through intelligent aids to decision making, procedure following, fault monitoring and diagnosis, etc. The increased flexibility available to cockpit designers as a result of these technological advances provides both opportunities and challenges. Significant improvements in capability, safety and efficiency appear to be possible, but they will not be easily achieved.

The AI field is presently the subject of much attention, and many claims are being made for its potential. Because of the diversity and richness of AI techniques, one can conceive of a wide range of possible applications in the area of intelligent aids for flight crew. However, it must be recognized that the design of AI systems is a subtle, multi-faceted art and that the technology is, relatively speaking, in a fairly early stage of development. Few, if any, AI systems have yet confronted the real-time operating constraints imposed by many flight deck tasks. Furthermore, the systems developed thus far have not had to operate in an environment where human factors considerations take on the importance they have on the flight deck. Consequently, the successful exploitation of AI technologies for flight deck aiding will require careful consideration of what crew tasks and/or functions should be aided and how that aiding should be implemented to maximize performance and acceptability. To avoid many pitfalls and false starts, it is also necessary to have a realistic assessment of relevant AI technologies both with respect to current capabilities and reasonable future expectations.

This report presents the results of a study aimed at developing a basis for applying AI to the flight deck environment of commercial transport aircraft. In particular, the study was comprised of four tasks:

1. Analysis of flight crew tasks .

2. Survey of the state-of-the-art of relevant AI areas.

3. Identification of human factors issues relevant to intelligent cockpit aids.

4. Identification of AI areas requiring further research.

The report is organized as follows. Chapter 2 contains the results for tasks 1 and 3. Structured interviews of experienced pilots was the principal tool used to analyze crew tasks to identify those that might benefit substantially from the introduction of intelligent aids and to uncover important human factors issues. In Chapter 2, the interview process and materials are discussed along with results of those interviews. Then, potential areas for intelligent aiding are identified along with relevant crew concerns and needs.

The state-of-the-art in AI, particularly those aspects that relate to automation and intelligent aiding, is surveyed in Chapter 3 and in Appendix A. The specific areas considered are expert systems, knowledge representation, planning, natural language, speech and AI tools. Chapter 3 contains a brief summary of the state-of-the-art in those areas; Appendix A contains more detailed review information.

Based on the needs and concerns brought out in Chapter 2 and the state-of-the-art of AI as discussed in Chapter 3 and Appendix A, research areas in AI that should be addressed for cockpit aiding applications are identified in Chapter 4. These research areas include many general and fundamental problems in AI as well as some more specific areas that emerge from the analysis.

Finally, Chapter 5 contains brief concluding remarks.

Acknowledgment

Many colleagues at BBN have contributed to this project and report. Dr. Walter Reitman and Dr. Robert Schudy participated in some of the pilot interviewing. Dr. Schudy also contributed significantly to the identification of relevant AI research areas. In addition, we have drawn heavily on a state-of-the-art survey of AI conducted by BBN staff for Dr. Kenneth Boff of AFAMRL under sub-contract to MacAuley Brown, Inc. That survey was edited by Dr. Ralph Weisedel and included contributions from the following: Expert Systems - Drs. N. Sridharan and A. Stevens; Knowledge Representation - Mr. J. Schmolze; Planning - Dr. N. Sridharan; Natural Language - Dr. R. Scha; Speech - Drs. M. Krasner and J. Wolf; and AI-Tools and Environments - Messrs. J. Gibbons and D. Allen.

We also which to acknowledge the help of Ms. Kathy Abbott of NASA-Langley in arranging for pilots to be interviewed and in providing guidance throughout the project.

Finally, we wish to thank the pilots, Capt. James MacIntyre (TWA), Mr. Lee Person (NASA), Capt. Dave Simmon (UA), and Capt. Jack Quigley (UA), who dealt patiently with our questions and provided many helpful insights into the needs and requirements for intelligent aids.

2. ANALYSIS OF FLIGHT CREW TASKS

Brief review of earlier efforts to identify areas in the commercial flight regime where provision of intelligent aids might enhance crew performance suggested that formulation of an adequate perspective required consideration of a large number of questions, among which are the following:

1. In what phases of flight are aids needed?

2. Where, within those phases, are aids most likely to be of benefit? What tasks of the crew/system within phases could be performed more successfully and with decreased workload as a result of the introduction of AI-based aids?

3. What aids already exist within each of those phases that might need to be eliminated or redesigned in order to ensure satisfactory crew/system interfaces with prospective aids?

4. What are the implications for crew workload in flight operations not directly addressed by prospective aids? If, for example, one proposed to substitute new, workload-reducing aids to navigation for those currently employed during the high altitude cruise phase, might a requirement to employ these during descent and approach phases actually have adverse impact?

5. What might be an appropriate distribution of priorities for research and development of prospective aids? This is clearly a difficult question when one considers that an ideal approach to integration of new technologies within the cockpit must necessarily take into account not only the objective requirements of the situation but also the subjective opinions of the crews who would need to be trained to use the technologies.

6. What human factors issues arise in connection with the introduction of such aids that will impact significantly on crew performance and acceptance.

It is clear from these questions that, in order to form a sound basis for design and development of AI-based aids, one must be able to characterize fairly accurately those aspects of crew/system interaction which demand high levels of attention and add significantly to workload. Further, since cockpit designs evolve over time and gradual transitions between old and new technologies must be anticipated, it is important to pinpoint functional aspects of crew performance where newly-introduced aids might have to interface with existing aids.

Our purposes during the analytic portion of this study were to acquire and to evaluate information relating to crew/system interaction and workload in the current generation of commercial aircraft as a means for identifying human factors issues critical to the design of intelligent aids. In order to conduct as broad an analysis as possible within the scope of the effort, we decided to pursue an approach based on structured interviews with experienced pilots. Response forms were developed to elicit answers to specific questions through a combination of pilot ratings on specific dimensions of workload and benefit, and pilot commentary obtained during intensive interviews. Below we discuss the methods used and the results obtained from the interviews.

2.1 Method

2.1.1 Preliminary Categorization of Crew Functions and Flight Phases

We began by specifying two lists. The lists were generated with the help of two senior airline pilots, whose extended discussions, along with those of two other pilots, later provided input to the consideration of possible AI-based aids. Brief discussions of the lists are presented below.

List 1. Systems with which Crew Interacts During Management and Control of Flight. Our purpose here was to obtain a generic list of the systems and subsystems with which a crew interacts significantly and repeatedly in the course of a normal flight. In particular, we were interested in obtaining items which required management, monitoring and control in order to satisfy given flight objectives. The final set of systems/subsystems agreed upon by the pilots is as follows:

1. Navigation
2. Hydraulic
3. Electrical
4. Flight Control
5. External Environment (other aircraft, runway, etc.)
6. Fuel
7. Thrust
8. Communications
9. Internal Environment (cabin pressure and temperature etc.)

List 2. Major Phases of Flight. Here, the total flight of a commercial aircraft was divided into phases that placed distinctly different demands on the functioning of the crew with respect to the nature of the activities that were performed and/or the accuracy of performance required. The list of phases that proved most acceptable to the pilots is one commonly used to categorize flight operations[1]

1. Pre-flight
2. Taxi out
3. Take off
4. Climb to cruise altitude
5. Cruise
6. Initial descent (to FL180)
7. Final descent (FL180-FL100)
8. Initial approach (FL100-LOM)
9. Final approach (LOM-TD)
10. Rollout
11. Taxi in

2.1.2 Response forms

Lists 1 and 2 provided the bases for development of response forms which could be given to subject pilots in order to obtain general information and to guide interviews related to three major questions:

1. "What workload is associated with each combination of activity and flight phase?" As an aid to dimensionalizing the admittedly difficult concept of "workload" effectively, pilots were instructed to consider the following while making their judgments:

   o What percentage of the total time available is spent performing tasks related to the activity in question?

   o Does the activity require cooperation among members of the crew or is it accomplished by one crew member?

   o Does the activity require continuous attention or can it be interleaved with other tasks?

---

[1] Post-flight procedures were eliminated from the list after pilots agreed that activities in that phase were not intrinsically different from those covered during earlier phases.

2. "With respect to what combinations of activity and phase do aids to performance already exist in the cockpit?" As a further refinement here, subjects were asked to estimate the degree of aid provided. Although not of direct interest to this study, classical aids such as autopilots and anti-skid systems were included in these estimates, as well as more innovative aids such as the fault monitoring systems in Boeing 757/767 aircraft.

3. "With respect to which combinations of activity and phase would the provision of aids be most beneficial?"

2.1.3 Personnel and Procedures

Our study employed the services of three senior airline Captains, one from TWA and two from United Airlines, and of a test pilot based at NASA Langley Research Center, VA. The commercial pilots were intimately familiar with systems and procedures for 727 aircraft, one had operational experience in the L1011 and 747, another had operational experience in the DC-10 and in the 757/767 series. In addition to their regular responsibilities, individuals in the group had taken on additional tasks on behalf of their airlines in areas of accident investigation and flight check/instruction. One was receiving instruction in space shuttle operations at Johnson Space Flight Center.

The Langley pilot had acquired much experience with experimental systems in his role as test pilot for NASA and, as a result of this experience, was able to provide an important perspective on possible guidelines for the development of AI-based aids.

As suggested in connection with the development of the Activity and Flight Phase lists, our initial intention was to require all pilots to provide scaled estimates on each of the three response forms while, at the same time, providing extended discussion of the rationale for each estimate. However, in an effort to extract as much information as possible in the time available, we soon departed from this intention and tailored our procedures to take maximum advantage of the background of each individual pilot. By the end of the data gathering process, one pilot had completed the entire set of three forms and provided many hours of additional discussion[2] ; a second had completed the Workload/Activity form and provided extended discussion as a substitute for the other two forms; the two remaining pilots had provided only discussion. Although such an approach clearly presented a compromise in the extent to which the accumulated data might otherwise have been treated quantitatively, it represented, for us, a substantial gain in the degree to which we were able to pursue significant comments made by the pilots during the interviews.

Beyond maintaining focus on points of interest to the project, we sought not to constrain the conversations with the pilots. This included not "building in" ahead of time, to either the response forms or the discussions, any specific assumptions regarding reliabilities and validities of prospective aids or airline-specific operating procedures. However, clarification proved to be necessary on both of these points: (1)The two pilots who completed Form 1 suggested that their answers with respect to workload during approach would vary somewhat depending on whether visual or coupled approaches were assumed. We encouraged them to respond differentially, as necessary . (2) All pilots speculated, at least indirectly, on the probable reliabilities and validities that would be necessary for AI-based system acceptability. This speculation was addressed by suggesting to the pilots that they assume the systems would, on these dimensions, exhibit approximately the performance of a modern autopilot.

Each of the four pilot interview sessions lasted from two to two and one-half hours. One of the sessions (JM) was conducted at BBN; the remaining three sessions were conducted at Langley.

The interviews were conducted under the supervision of an individual with a background in human engineering and systems analysis. This individual was familiar with techniques for obtaining verbal protocols, having been involved previously in a number of such processes for other complex systems. He also had a background in examination of human factors issues in commercial aviation. In addition, there was also present for the interviews an individual who was familiar with AI and could suggest and/or scope what might be reasonable technically. This was important for steering, and keeping, the discussion on appropriate paths.

---

[2] This captain was a consultant to BBN during early phases of the project and provided much additional perspective on procedures, systems, and potential acceptability of new crew aids.

2.1.4 Limitations of the Analysis

Because of the scope of the effort, the interview process and the definition of aiding possibilities was limited in certain ways. First, the discussions were limited primarily to present operations. This is a significant constraint in that a major need for additional aids is likely to arise as a result of future changes in the air traffic system and environment, such as more demanding approach paths, etc.

Second the response forms and, with minor exceptions, the discussions, centered on normal operations, although abnormal and emergency situations are logical areas for which to consider advanced AI-based aids. To have addressed these areas well required, in our view, highly detailed and focused questions that were beyond the scope of this effort. Nonetheless, some of the content of the interviews was relevant to abnormal and/or emergency conditions and this was duly noted.

Finally, it should be recognized that there is insufficient quantitative information to develop conclusions based on statistically reliable data. None of the results by itself clearly establishes the need or lack of need for new performance aiding techniques in the cockpit. This is particularly true of information collected on the response forms, which were utilized primarily as a mechanism to guide discussion. It is, rather, in the aggregate of findings that suggestions regarding the nature and benefit of such techniques can be found.

2.2 Results

In sub-sections 2.2.1–2.2.4 below, we present discussions of the findings associated with the response forms completed by two of the pilots. The findings which grew out of the interviews conducted with these and the remaining two pilots are presented in subsection 2.2.5.

2.2.1 Question 1: What Workload is Associated with the Various Combinations of Flight Phase and Crew Activity

This form was completed by two captains (Capt's. "A" and "B" with the results shown in Table I and Table II below. For purposes of more quantitative assessment, each of the responses has been assigned a score between 5 (very high) and 1 (very low). The means of each row and each column and their corresponding ranks are shown on the tables in order to provide indications of the distributions of workload among phases and among activities.

Table I. Responses of Capt. A to Workload Question

SYSTEM	Nav.	Elect.	Hydr.	Flt.Ctrl.	Ext. Env.	Fuel	Comm.	Thrust	Int.Env.	MEAN	RANK
PHASE											
Pre-Taxi	5	3	3	1	2	4	3	2	3	2.89	5
Taxi Out	2	1	2	2	5	2	5	2	3	2.67	8
Take Off	1	1	4	4	4	2	2	4	1	2.56	9
Climb											
Initial	4	1	2	4	4	2	4	3	1	2.78	7
Enroute	2	1	2	3	3	3	4	2.5	2	2.50	10
Cruise	4	1	1	2	2	3	3	2.5	2.5	2.33	11
Descend											
to FL 180	4	1	1	3	4	3	4	4	4	3.11	2
to FL 100	4	1	1	3	4	3	4	4	3	3.00	3
Approach											
to LOM	5	2.5	4	4	4	2	4	3.5	3	3.56	1
to TD	3	2	2	3	3	2	3	3.5	4	2.83	6
Rollout	1	1	3	3	5	1	1	4	2	2.33	11
Taxi In	3	2	4	2	5	2	4	1	3.5	2.94	4
MEAN	3.17	1.46	2.42	2.83	3.75	2.42	3.42	3.00	2.67		
RANK	3	8	7	5	1	7	2	4	6		

KEY:
- 5 = very high
- 4 = high
- 3 = moderate
- 2 = low
- 1 = very low

Table II. Responses of Capt. B to Workload Question

SYSTEM / PHASE	Nav.	Elect.	Hydr.	Flt.Ctrl.	Ext. Env.	Fuel	Comm.	Thrust	Int.Env.	MEAN	RANK
Pre-Taxi	4	1	1	1	1	1	1	1	1	1.33	8
Taxi Out	4	1	3	1	5	2	5	2	2,5	2.83	2
Take Off	3,5	1	3,5	5	4	1	4,5	2,5	1	2.89	1
Climb											
Initial	4	1	1	4	4	2,5	4	3	2	2.83	2
Enroute	2,5	1,5	2	3	3	2,5	2,5	2	2	2.33	6
Cruise	4	2	1	2	2	3	2,5	2	2	2.28	7
Descend											
to FL 180	4	2	2	3	2,5	3	3,5	2	2,5	2.72	3
to FL 100	4	2	2	3	2,5	3	3,5	2	2,5	2.72	3
Approach											
to LOM	3,5	2	1	3,5	4	3	4,5	2	2,5	2.89	1
to TD	3	1	1	3,5	4	2	4	2	3	2.61	5
Rollout	2	2	3,5	3,5	4	1	3,5	2,5	2,5	2.72	4
Taxi In	2,5	2	4	3,5	5	1	3,5	2	2,5	2.89	1
MEAN	3.42	1.54	2.08	3.00	3.42	2.08	3.50	2.08	2.17		
RANK	3	7	6	4	2	6	1	6	5		

KEY:
5 = very high
4 = high
3 = moderate
2 = low
1 = very low

The ratings presented in Table I and Table II with respect to workload across activities are in substantial agreement ($r_s > 0.90$) and suggest that, from the perspective of these pilots, workloads associated with navigation, communication's management, and external data management are the three most demanding activities when averaged over all phases of flight.

Comparison of the right hand columns of Table I and Table II suggests a low correlation ($r_s < 0.10$) between the pilots' judgements of relative workload among flight phases. An obvious question arises as to why this might be so, particularly in view of the high degree of agreement among rankings of flight activities.

There are at least two factors that could explain the outcome. One is a procedural artifact that could result from the fact that the pilots were requested to complete their ratings for a given flight phase before considering the next phase. This would tend to ensure that anchor points in their judgement processes were less variable within phases than between phases and could lead to greater agreement. The second possibility is that intrinsic differences among airline procedures, aircraft and route structures contribute to greater variety in relative workloads across phases than that associated directly with management and utilization of systems.

Despite the low correlation, the pilots seem agreed that descent phases and the initial approach phase produce high levels of workload. They are also in general agreement that "En route Climb" and "Cruise" generate relatively low workloads under normal circumstances.

2.2.2 <u>Question 2:</u> What Aids Currently Exist in the Cockpit

The form associated with this question was completed by Capt. A with the results shown in Table III. Cells containing the letter "A" (absent) indicate combinations of system activity and flight phase where, in the pilot's opinion, there are few, if any, salient aids. Cells containing the letter "P" indicate combinations where such aids are present. Numbers associated with "P"'s indicate specific types of aids considered and are identified in the Key to the Table. A "P" unaccompanied by a number indicates an assist to the activity that is rendered by a combination of systems/devices, no one of which is particularly salient.

It should be noted that the aids cited here are characteristic of 727, L1011, and 747 vintage aircraft.

Table III. Responses of Capt. A to Current Aids Question

SYSTEM / PHASE	Nav.	Elect.	Hydr.	Flt.Ctrl.	Ext. Env.	Fuel	Comm.	Thrust	Int.Env.
Pre-Taxi	A	A	A	A	A	A	A	A	A
Taxi Out	A	P1	P	A	A	P	A	A	A
Take Off	P2	P1	A	A	A	A	A	A	A
Climb									
Initial	P2	P1	A	P2	A	A	A	A	P
Enroute	P2	P1	A	P2	A	A	A	A	P
Cruise	A	P1	A	P2	A	A	P3	P4	P
Descend									
to FL 180	A	P1	A	P2	A	A	P3	P4	P
to FL 100	A	P1	A	P2	A	A	P3	P4	P
Approach									
to LOM	A	P1	A	P2	A	A	P3	P4	P
to TD	A	P1	A	P2	A	A	A	P4	P
Rollout	A	P1	P5	A	A	A	A	A	A
Taxi In	A	A	A	A	A	A	A	A	A

KEY:
- A = Significant Aids Currently Absent
- P1 = Aid Present: Auxiliary Power Unit (APU)
- P2 = Aid Present: Auto Pilot/Flight Director
- P3 = Aid Present: Select Call (SELCAL)
- P4 = Aid Present: Auto Thrust
- P5 = Aid Present: Anti-Skid

Collectively, they may or may not present an accurate picture of the level of assistance available in current designs. To the extent that they do portray such a picture, they represent technologies with which a newer generation of AI-based aids may need to interface during early stages of development.

In general, these results suggest that few aids exist for management of fuel and hydraulic systems, for acquisition and processing of data in the external environment, and for navigation.[3] The lack of aids may, moreover, be most apparent in "Pre-Taxi", "Rollout" and "Taxi-In" phases of flight.

2.2.3 Question 3: Where Would Aids be of Benefit?

The responses of Capt. A to the question of where provision of new or additional aids would be of benefit are presented in Table IV. Cells containing diagonal lines indicate combinations of activity and phase where such provision was judged to be beneficial. Certain of these cells also contain the letter "A". These indicate combinations in which aids were said to be lacking in response to Question 2.

The results in the table suggest that further aids to navigation and to communications are needed throughout all phases of flight. It also suggests that aids related to fuel and thrust management, navigation, communication, and monitoring of nearby traffic would be of significant benefit.

2.2.4 System Activities/Flight Phases Associated with High Workload and In Need of Further Aiding

The responses of Capt.'s A and B to Question 1 and of Capt. A to Question 3 deal separately with distributions of workload and potential benefit. It is of considerable further interest to determine what combinations of system activities and flight phases might be associated with high workload <u>and</u> high benefit, since these combinations could provide initial foci for research and development of AI-based aids. Such a determination has been made for the pilot who responded to both questions and is presented below in Table V. An "X" in a cell indicates a

---

[3] Considerable aiding has, of course, been introduced into areas of fuel, hydraulic, and electrical systems monitoring and reporting and into navigation in, for example the 757/767 series.

Table IV. Responses of Capt. A to Further Aids Question

SYSTEM PHASE	Nav.	Elect.	Hydr.	Flt.Ctrl.	Ext. Env.	Fuel	Comm.	Thrust	Int.Env.
Pre-Taxi	///A////								
Taxi Out					///A////		///A////		
Take Off					///A////			///A////	
Climb									
Initial	////////						///A////		
Enroute							///A////		
Cruise	///A///								
Descend									
to FL 180	///A////						/////////	/////////	
to FL 100	///A////						//////////	/////////	
Approach									
to LOM	///A////		///A////		///A////		/////////	/////////	
to TD								/////////	/////////
Rollout									
Taxi In							///A////		///A////

KEY:
///A/// = Aids Currently Absent and Highly Desirable
/////// = Aids Currently Present but Additional Aiding Highly Desirable

Table V. System Activities/Flight Phases Associated with High Workload That May Benefit From Introduction of Aids. (Based on responses of Capt. A)

SYSTEM / PHASE	Nav.	Elect.	Hydr.	Flt.Ctrl.	Ext. Env.	Fuel	Comm.	Thrust	Int.Env.
Pre-Taxi	X								
Taxi Out					X		X		
Take Off					X			X	
Climb									
Initial	X						X		
Enroute							X		
Cruise	X								
Descend									
to FL 180	X						X	X	
to FL 100	X						X	X	
Approach									
to LOM	X		X		X		X	X	
to TD								X	X
Rollout									
Taxi In							X		X

combination of activity and phase that was judged high or very high in workload (VH or H in Table 1) and a candidate for aiding (////// or ///A/// in Table IV).

The results in Table V suggest that efforts aimed at developing aids to the accomplishment of navigation, communications and thrust management activities could have high payoff across all phases of flight. Also, they suggest the high utility of aids to acquisition, processing and management of external environmental information, particularly during initial and final phases.

2.2.5 Pilot Commentary

The results obtained with the response forms provide indications of where workload is high and of where there may be benefit, in terms of increased performance accuracy and/or a decrease in workload, that would accrue to the introduction of various aids. They do not, of course, suggest the natures of the problems that need to be addressed. Nor do they suggest the types of technologies or applications that might address the problems. As indicated earlier in this chapter, information regarding these factors was obtained through interview.

The material presented in this section consists primarily of statements made by the pilots in response to specific questions posed by the interviewers or as parts of longer dialogs aimed at explication of current operating procedures, current problems and/or prospective aids. The material is organized into seven subsections. The first four of these contain statements relating to aiding of specific functional activities: "Navigation", "Communications", "Fuel/Thrust Management", and "External Environment". The remaining three subsections present additional statements relating to "Fault Monitoring", "Control Responsibility", and to the general concept of a "Pilot's Associate".

2.2.5.1 Navigation

It is frequently the case that the aircraft is forced to deviate from its normal flight plan. Aids that could facilitate development of new or modified plans and that could help satisfy various specific goals (e.g., maximize fuel savings, maintain schedules as closely as possible, etc.) are desired. Such aids would also be important in abnormal situations where re-routing and other factors necessitate changes in plans.

The manner in which such aids are implemented--what inputs are required, how and when they are to be made, and what is required to verify them--are important human factors considerations. The following comments provide a sense of the concerns.

> "...Flight plans -- routing, etc. --always change right after you take off. It would be a big advantage if inputs from all the button pushing that goes on now in the airplane and in ATC could be integrated and used for nav/fuel expenditure, flight profile planning and replanning, etc. -- (probably in the cards.)"

> "...The problem that was occurring .. was trying to restring a flight plan or an approach plan in the computer, and (the crew under discussion) flew right past the airport trying to set up an approach. So that becomes an operational thing that you can say, well, if everything starts to go to pot, you can go back to Plan A. You don't have to use Plan C all the time. (There) should be a simple way of doing it, but, in fact, it appears not to be as simple as I thought it was, because apparently they're going back to trying to restring the thing while right here on approach, feeling like they have to get it into the computer."

> "...Well, we started it (telling our guys that once they got into the terminal area, they shouldn't try to reprogram) .. almost every other airline went along .. In fact, TWA wanted to bolt a plate over (the keyboard) during the approach. (A smart system could say, "We're on approach; you're not supposed to be playing with this.) .. This is where you could turn what could be a negative automation into a positive (automation), by not letting him do it."

2.2.5.2 Communications

Communications in the cockpit represent a major source of workload (and sometimes confusion). Relatively simple systems that could aid in, for example, frequency selection would be helpful. More complex systems that could monitor incoming communications and information that would normally be on a headset and then filter it to present only information desired at the time--a difficult technical feat--would be of extremely high value.

> "...If I were having an assistant .. I would have, say, a big black button out there that said, "inhibit incoming calls, (commence) final check". (Then I could) do the final check, then it could give (me) those calls.

In other words, "DON'T BOTHER ME RIGHT NOW -- I've got something more important .. This would be an area where you would want different levels of communications or filtering."

"...If you're taxiing out, you have an interest as to what number you are, which way you're taking off, weather, problems, .. you want to know what the guys in front of you are doing, and what the thinking is, and that helps you make the same kind of a decision."

"...Turning on ground control early in the morning (during pre-taxi) is an incessant line of chatter. Sometimes you have to wait 4-6 minutes just to be able to get that fraction of a second break where no one is talking. You grab the microphone and hope that the rest of the guys didn't cut you off once you started to talk. It would be really nice if you could do it through the ACAWS."

"...The tower has cleared somebody to land on runway 33 and he's clearing me for runway 36 and (I) look out there and say,"that's far enough away, go ahead and go." At that point, I'd like to hear two airplane communications. I'd like to know what's going on .. as far as my routing is concerned, I don't care. Nobody is really interested in that and I'm not interested in what they're doing."

2.2.5.3 Fuel Expenditure Planning and Thrust Management

It was our initial intention to include in the "Thrust Management" category only those pilot/crew actions related directly to throttle control, and to include in "Fuel Management" activities more significant cognitive requirements, such as planning of fuel resource expenditures, monitoring of resources against plan, and replanning where necessary. This distinction proved to be a major source of difficulty to the pilots during completion of the workload response form (see Tables I and II) and impossible to maintain during discussion, particularly when the pilots began to reflect on possibilities for combining AI-based planning systems with auto-throttle control. The two categories are merged into a single category in the comments below.

"An artificial intelligence system would say, "Hey, this is just what (you) did the last time. (You) really screwed up last time. (You) thought it was heavy on the left--(You) burned the fuel out of the left. (In fact) it was heavy on the right. Here it is on the next flight, looks exactly the same as it did the last time. (Will you) just go ahead and do it wrong again."

"...one of the biggest things in the airplanes now is fuel management--you gotta save fuel..almost more important than safety. ATC is gonna bring you in high and drop you fast, and it all sounds good, and then you gotta make last minute changes or get in line and the whole thing (conservation) goes out the window."

"...Anything in fuel or fuel efficiency and working out calculations and numbers and things like that would come in as a high priority item..we spend 1.3 billion a year (on fuel)."

"(ATC is) trying to do things that they cannot do reasonably...there's not a weatherman in the world that has the kind of accuracy (to predict thunderstorms at a great distance so) you back up traffic, you miss connections...they start people down 80, 150, 200 miles early. Fuel flow goes sky high when you do that."

2.2.5.4 Acquisition and Management of Environmental Information

This is a safety-critical, high workload area for the crew. Systems that would monitor sensors and other data and utilize already-stored knowledge of routes, schedules, etc. to aid the crew in avoiding collisions with terrain or other aircraft would be desirable. Obviously, such systems would have to outperform current systems with respect to detection, computation, display capabilities and false alarm rate.

"...External data...I think that's probably at least high in that area (Initial Climb). You are listening and trying to be aware of other things going on, who took off next to you, which way they're going, ATC clearance, looking out for things like windshear, windsocks, and things like that, so there's a lot that's going into that."

"...anytime you're climbing you're still looking and listening, not so much for communications but there's still the need for external data--listening, looking and being aware of traffic. Anytime, they're gotta be high."

"...If you start going into weather, then the need for external data becomes high."

2.2.5.5 Fault Monitoring and Diagnosis

The continued development and utilization of fault monitoring and diagnosis systems were judged by all the pilots to be highly desirable. The capabilities of some of the sophisticated systems which have already made their appearance in the cockpit, and operate within the context of rule-based production systems, were regarded to be very attractive. Within this broad area, engine system monitoring was cited as a prime candidate. The desire is for systems that can answer the following types of questions in timely, reliable and understandable ways:

1. What systems have failed?

2. What systems are likely to fail in the near future?

3. What capabilities remain?

> "...the first place I would like to see artificial intelligence is in the engine monitoring end of the airplane--engine systems monitoring."

> "...If the system is smart, it will say "oil pressure 10 psi over the last 5 minutes". It will start to look for temperature, quantity, and will direct me as to what to do with the engine. That is what I think a smart flight engineer would be doing..I'm not interested in a box that sits there and waits until all of the oil's gone and the pressure goes to zero.."

> "...I would want (the system) to know when to tell me (well before values approach limits): "At the present rate of temperature increase in your hydraulic fluid, you're going to exceed the limit in five minutes.""

> "...It seems to me that artificial intelligence, rather than concentrating on areas of flying the airplane--that's not the area that would be most beneficial--at least to start with, would be more systems oriented to present information in a clear way to the pilot to be able to answer a lot of "what if" questions which a computer can do very well and very fast. Like you've got an electrical problem: You lost a couple of busses and a couple of subsystems. What do you have left? What happens if other things happen down the line? What are the tie- ins to the system? How does losing a couple of electrical busses affect not only the electrical system but the hydraulic, and the fuel system and a number of things like that. This kind of

information is what artificial intelligence can do fairly well for us. And fairly quickly."

2.2.5.6 The Concept of a Pilot's Associate

One of the more intriguing ideas to arise during the interviews and one which provoked much discussion concerned the possibility of developing an AI-based system that could provide support to management and control functions of the Captain in much the same way that a First and/or Second officer currently does. In some circumstances, this "Pilot's Associate", as it has been called by the Defense Advanced Research Projects Agency (DARPA) in connection with military aviation, might substitute briefly for a busy or otherwise incapacitated crew member, while in others, it might permanently replace him/her. The idea of developing and utilizing such a system in the commercial cockpit is attractive for at least three reasons:

1. There is a significant trend toward smaller crews in the design of new commercial aircraft. The philosophy represented in the design of 757/767 crew stations and procedures serves as an example of what it may be reasonable to plan for in the way of crew complement and division of responsibility. The move to smaller crews may result in increased workload for one or more remaining crew. Any system that could aid in offsetting that increased workload might be highly desirable. It might be particularly desirable if it could sense when one or more members of the crew was or could become overloaded and then assume part of the load.

2. As the pilot's role as a supervisor and monitor of automatic systems continues to evolve and he/she is required to provide less and less direct input to rudimentary flight control functions, his/her immediate "feel" for environmental forcing functions and the compliance of the aircraft may decrease substantially — well below that which exists in current systems. This could become a serious problem when rapid takeover of control by the pilot is necessary. An "Associate" might be able to smooth this transition.

3. During abnormal or emergency conditions, it would often be useful to have access to the kind of expert knowledge and guidance that is currently available in the heads of designers of the aircraft and its systems and subsystems. If a system could be designed with such experts' knowledge bases and the capacity for symbiotic interaction, the crew's performance might be

more effective that it would be if it were merely based on standardized procedures.

It was clear from the discussions that the concept of a Pilot's Associate would have to be sharpened considerably and that important questions would need to be addressed before the scoping of such an aid could realistically proceed, The following (not un-related) questions provide examples of some of the deeper issues requiring consideration and also illustrate that functionality and human factors interact strongly.

1. Should the Associate merely replace the "missing" crew member? If, for example, the responsibilities of the Second Officer (Flight Engineer) are to be incorporated into the task structure of the First Officer or, more generally, the Pilot-Not-Flying, does the Associate provide aid only to satisfaction of those inherited responsibilities?

2. Should the Associate "think" or does it merely detect and report? Does it need to be endowed with inferential and decision making capabilities, or is it sufficient that it act as a sophisticated monitor of events and repository of recommended procedures? What are the limits of these inferential and decision making processes--should they attempt to emulate only rule-based behavior or include knowledge-based behavior as well?

3. If the Associate is required to exhibit knowledge based behavior and to interact with the crew, whose knowledge base should it obtain? Should it contain the accumulated knowledge and expertise of the Pilot or the First Officer or both? Or does it need to have the capacity to make inputs based on, for example, engineering knowledge that is not normally available to any crew member? Are there circumstances in which it might need to emulate group problem solving behavior and, therefore, perform in the aggregate like an expert panel of designers and operators?

4. Should the Associate act as well as think? Should it be endowed with cognitive skills and effectors appropriate to the performance of aircraft management? Under what circumstances should such skills be enabled--when it senses the need?, when told by the crew?

As indicated, there was much discussion of the proposal to develop such an "associate". The following quotes provide some of the flavor of this discussion.

"...the nicest thing in the world..would be to have an (AI-based) copilot who is absolutely knowledgeable about all the procedures and all the airplane's capabilities and to have a flight engineer who is absolutely knowledgeable about all of the basic aircraft systems. You think about these people being 100% dedicated--flight engineer just scans from instrument to instrument to to instrument. He knows what the normal range should be, he knows what it was on the last flight, he knows if there's been any change in the last 15 minutes, and anytime there's anything I should be aware of, he gives me that information."

"...My co-pilot would know everything there was to know about the airplane's performance envelope and he has all the sensors to know what the wind is doing and what the temperature is doing, what the airplanes gross weight is, what the best mach number would be and what the best altitude should be for the weight and the mach number, and he advises me of those thing."

"..Remember, my crew chief knows every piece of wire, etc., and although he may not pass that information to me, he may have completed analyzed everything that's going on in that airplane and jotted it down, and when he gets on the ground, he gives that to Charlie and says, "fix it"."

"...I'd lean over here to "Charlie" and I say, "Charlie, I'm going to watch the airplane and I want you to get our display and I'll take a look at it and tell you whether or not I like the way it looks..now I can either fly it myself..or I can turn it over to the automatic."

2.2.5.7 Control Responsibility

Perhaps <u>the</u> major human factors concern of the pilots in regard to introduction of AI-based aids in the cockpit is that, in some circumstances, operations with such aids may leave critically unanswered question, "Who is in control"? There were two critical dimensions to this concern:

1. AI-based systems may take away control initiatives.

2. AI-based systems are likely to be complex and difficult to understand. They may draw on data that is largely unobserved by the pilot/crew. For this reason, they may evaluate and choose among alternatives whose characteristics and utilities are not directly

available to the pilot/crew. Given time and workload constraints, the pilot/crew may not be able to approve system choices before implementation is required.

The concern is legitimate. Unlike current autopilots and fault detection/reporting systems, which operate on "simple" inputs, generate easily verifiable outputs and produce predictable consequences when disabled, AI-based systems have the potential for quickly manipulating large quantities of diverse data, executing rule-and knowledge-based algorithms beyond the immediate comprehension of the pilot/crew, and effecting significant and unseen changes in remote aircraft subsystems. These systems must, of course, be designed in ways that allow the pilot/crew to remain "in the loop". How to accomplish that remains a difficult problem to which the pilots in these interviews were very sensitive.

"...the worst thing is to see something happen and not know what is going on. That puts you in a confused loop. You don't know what is happening or why, and now you're spending your time trying to think about why did (the AI do) something in the first place that you didn't want to happen or, at least, didn't understand the reasons for in the first place. Even though it (may have been) the right thing to do, you didn't want it to happen. So now you're confused and you're wondering what it will do now."

"...Everytime you do something different from an automated standpoint, the pilot has to not only know how to do it himself manually, which he doesn't practice as much, but then he has to spot faults with the automatic system and know when to take it out of the loop, and that's got to be part of the same kind of process".

"...(In reference to a system that informs the pilot that some condition may require eventual shutdown of a subsystem) "that's good .. that's excellent. Because then it becomes advisory, and even in a case, of a light is going to come on and say, "You have an engine fire in #2 engine", and say it out loud, ok, in that case,; or if you want to have it (present) some kind of alerting device that wakes the guy up if he happens to be asleep (unlikely?)...But somehow, before I would want something to just automatically shut it down, I would want a system that would maybe say, "In 10 seconds engine #2 will be shutdown". And then if I'm in critical situation, I can reach up and disconnect the (automatic shutdown sequence) -- no. I don't want (that to happen) -- let it (continue)."

> "...you have to have some way of interacting. (The system) doesn't just say, "I did this". It's got to say, "I did this, because. You've got to let me (,the system,) know you (,the pilot,) know I did this ... (and) did this because you were busy doing something else). ...this is all part of keeping the man in the loop."

2.2.6 Human Engineering Considerations

A host of questions related to human engineering of system/crew interfaces arise in connection with the development of AI technology. Many of these are classical questions, but they take on added significance in the light of expanding technological opportunities for information acquisition, processing and display. The following two are representative:

1. What methods/techniques of information input are most consistent with goals of rapid, accurate and reliable performance by the crew? Which provide the lowest average increase in workload?

Obvious alternatives are:

- o keyboards and function buttons
- o voice input
- o touch-sensitive displays

2. What form(s) should the outputs of AI-based system processing take in order to assure rapid and accurate assimilation by the crew?

- o printed copy
- o synthesized speech
- o visual display (CRT)
- o auditory and visual alarms

Although the merits of technological alternatives associated with such questions cannot be judged without a reasonably specific concept of the functions(s) to be performed by an intelligent system, the general desirability/undesirability of certain of the alternatives was touched on during our discussions. These are summarized in subsections below along with our comments.

2.2.6.1 Keyboards and Function Buttons

The value of function buttons and keyboards as input devices is undeniable. Their signification is unequivocal and virtually instantaneous and their use conserves space. At the same time, they have at least three obvious disadvantages that are difficult to design away:

1. They present opportunities for human error, particularly during conditions of high workload. Such errors, when they occur, may not be immediately obvious. This is especially true when lengthy inputs are made via keyboard.

2. It is difficult, if not impossible, for the operator to accomplish other tasks while pressing buttons or keys. If the input task cannot be deferred, the pilot has the choice of accepting the risks that go with a division of attention, deferring the competing task(s) or adding an increment to the workload of another crew member. None of these alternatives may be acceptable in some circumstances.

3. Current cockpit layouts require that the operator remain head down while accomplishing most button and keyboard inputs. Again, depending on circumstances, the input task may be accomplished only at a cost in performance in other competing tasks. In this connection, our pilots recalled instances (See Section 2.2.5.1) where crews already well into the approach phase chose, probably inappropriately, to type revised coordinates into their automatic system rather than fly to the coordinates manually, thereby risking critical losses in awareness of external environmental factors. The context in which these events were recalled suggested a latent "trap", namely, the typical degree of success experienced with the automatic system may actually have prevented consideration of hand flown corrections to the flight path.

2.2.6.2 Voice Input

It may not be surprising that, despite increasing use of function buttons and keyboards for input of information (e.g., in navigation), one of the most desired modes for transmission of information under most circumstances is voice. It is the mode very likely to be preferred for communications both within the cockpit -- that is, between the crew and an AI-based system -- and for up-and-down-links.

The preferences of the pilots in this study for voice mode seem to have largely to do with the naturalness and efficiency with which voice communications can be carried on, and, perhaps more importantly, with the fact that voice transactions can be completed in parallel with the performance of ongoing control and monitoring functions.

The pilots in the group were aware of efforts to develop speech understanding technology and of some of the difficulties inherent in that enterprise. One noted, for example, that changes in the characteristics of the voice under stressful conditions might pose a severe problem. The responses suggested that a conversational mode mediated by natural language would be required for most crew-system interaction under any but the most routine circumstances.

It is important when assessing the potential utility of voice or synthetic speech as an input/output modes to consider that a virtue which verbal outputs may not share with, for example, a warning light or horn or even printed copy, is persistence. The warning light or horn typically remains on until the condition it signals is attended to. A printed message is somewhere in the cockpit until it is disposed of. But human memory for the content of spoken messages is typically short. If the full potential of systems that will employ voice or speech synthesis techniques is to be realized, methods which permit rapid pilot/crew access to the content of previous transmissions must be developed.

2.2.6.3 Touch Sensitive Displays

Although major disadvantages of touch-sensitive displays are that they, like buttons and keyboards, generally require the user to remain head-down and to concentrate carefully on his/her task, there may be an off-setting advantage in the application of this technology to input of navigational map-based information. Depending upon the scale(s) of the map(s) used and the accuracy of specification requirements for a given maneuver, one can imagine a crew member actually pointing at map locations rather than keying in coordinates in order to trigger route planning or re-planning computations by an intelligent system. Or, pointing and keying modes might be intermixed in such a way that global desires were initially imparted to the system via touch control panel and then final tuning of plans proposed by the system was accomplished via keyboard.

2.2.6.4 Printed Copy

Few direct comments were made during the interviews regarding the desirability/undesirability of printed copy outputs beyond those already generated in the cockpit by ACARS. One might guess, however, that the addition of more printed material to the already large body represented by maps/charts, procedure handbooks, etc., would not be well received by crews for at least two reasons:

1. Information presented in any form must be at least scanned before it is possible to conclude that it is relevant, partly relevant or not relevant. The scanning probably cannot be accomplished as efficiently with printed material as it can be with information delivered on a status board, on a graphic display, or auditorially. As such, additional critical time may be lost and/or more division of attention may be required with this method of presentation than with most others.

2. Printed copy can pose a storage and retrieval problem in the cockpit, that is not necessarily solved by the addition of more and better engineered space. Testimony to what may already be an accessibility problem is the retro-fitting by several airlines of large, spring-loaded paper clamps, suitable for holding approach charts and other important information, to the yokes of 727's.

2.2.6.5 Synthesized Speech

As a result of the growing use of speech synthesizers to generate commands (e.g., low altitude "PULL-UP...PULL-UP") and advisories, pilots have at least nodding acquaintance with the potential of this technology for providing human-like voice inputs to the crew. Largely for the same reasons that voice input is desirable--"naturalness", efficiency and allowance for parallel processing--the use of synthesized speech is generally preferred to other output modes.

2.2.6.6 Auditory and Visual Alarms

Simple auditory alarms, instrument flags and annunciator panels are, of course, the traditional methods for advising crews of out-of-tolerance conditions, equipment failures, etc., and most have the virtue of being easily understood <u>when attended to.</u> Their major disadvantages are that they are occasionally a nuisance and that they serve primarily an alerting function, not a directive function. When they become a nuisance, either because they divert attention from a critical task being performed, they report a condition that has already been appreciated, or they are thought to be false, they are often ignored and/or cancelled, with undesirable results.

While discussing these characteristics of current alarm systems, several pilots in this study also commented unfavorably on the increasing extent to which (particularly) auditory alarms are being incorporated into systems. One complained of having to attend "Sound School" so that he would be able to "sort out all that crap". He had the distinct feeling that the limits of human auditory discrimination capacity under stress were rapidly being approached.

3. STATE-OF-THE-ART OF ARTIFICIAL INTELLIGENCE

Six areas of AI research and development were identified as being of prime importance to intelligent cockpit aiding. These are:

- o Expert Systems
- o Knowledge Representation
- o Planning
- o Natural Language
- o Speech
- o AI Tools

The Expert Systems, and planning areas are particularly relevant to specific types of aids and to the crew associate concepts. Natural Language and speech are primarily of interest with respect to crew/system interface questions. Knowledge representation and AI tools are of significant interest because they are fundamental aspects of AI that impinge on all developments. These six sub-areas of AI have tended to function somewhat separately in the past. However, we expect that they will interact much more substantially and frequently in the future; certainly, the capabilities required for advanced intelligent aiding of the type desired on the flight deck cannot be attained without such interaction.

3.1 Expert Systems

We define an Expert System to be a computer program that explicitly incorporates knowledge based in significant part on symbolic representation of facts, rules of thumb, strategies, concepts and heuristics that an expert might use in solving one of a class of problems. Thus, it is a knowledge-based system. The knowledge is supplemented by an inference mechanism that enables drawing conclusions from the knowledge and generally, by some capability for the system to "explain" its decisions or recommendations. Expert Systems is clearly the most visible area of AI with it receiving an enormous amount of interest.

Some expert systems 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 is, however, 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. Finally, current systems have generally not had to function under the solution time constraints that would be characteristic of dynamic environments such as aircraft operation.

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

TABLE 1. SOME EXTENSIVELY DEVELOPED EXPERT SYSTEMS

Name	Application	Organization
ACE	Analysis of telephone cable trouble spots	Bell Laboratories Whippany, NJ
CADUCEUS	Internal medicine	Univ. of Pitts-burgh
CASNET	Consultation regarding glaucoma treatment	Rutgers University
DELTA (formerly CATS-1)	Troubleshooting diesel locomotives	General Electric
DENDRAL	Projecting molecular structure from mass spectrograms	Stanford Univ
DIPMETER ADVISOR	Interpreting oil well drilling log data	Schlum-berger -Doll
MDX	Medical diagnosis for cholestasis	Ohio State Univ.
MYCIN	Diagnosis and treatment of bacterial infections	Stanford Univ.

	of blood	
PROSPECTOR	Predicting likely ore deposits	SRI Int.
PUFF	Consultation regarding pulmonary disorders	Stanford University
STEAMER	Training regarding operating of a steam propulsion plant	BBN Labs.
XCON (formerly R1)	Configuring VAX computers given a customer order	Digital Equipment Corp.

As can be seen, a major application of expert systems, that is of particular interest in cockpit aiding, is fault monitoring and diagnosis. This is an active development area and systems are moving from pure rule-based designs to ones employing more sophisticated knowledge representation schemes that attempt to represent physical systems structure and function and admit more powerful (deep) reasoning.

Unfortunately, virtually all of the work in diagnosis to date has not been concerned with, or had to deal with, the kinds of issues imposed by operation in a commercial transport environment. In such an environment diagnosis must occur in extremely timely way and be very reliable. The diagnostic systems must also be able to "explain" their diagnoses quickly and clearly (it is not enough to trace the rules that have been invoked). The ability to conduct further tests to confirm or discard the diagnosis can be quite limited in this environment. Finally, it is often not identifying the fault that is important but, rather, reasoning about that capability remains. This kind of reasoning has not been an issue in past expert diagnostic systems.

Though a potentially large class of applications of current expert systems 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, particularly to the classes of applications indicated in Chapter 2. Very briefly, most of the systems that currently exist, and most of those under development, are rule-based systems and, as such, are based on a limited paradigm. They tend to function in very narrow domains and they have difficulty dealing with inferences that require knowledge of structural, casual, spatial and temporal relationships. They break down when knowledge is incomplete or inconsistent. Few, if any, of the systems run in real-time.

Finally, they tend to have only rudimentary explanation capabilities. In addition to these limitations of present systems, the effort to elicit knowledge and to build the systems (knowledge engineering) is currently very large.

This picture is likely to change to a substantial degree in the foreseeable future. Due to the high investment research in expert systems arising from the 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 for acquiring knowledge

The DARPA Strategic Computing effort will also contribute substantially to the next (second) generation of expert systems. In particular, it can be expected to 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 in that program, one should note particularly:

- o major increases in speed (this will help foster real-time expert systems)
- o advances in automating the acquisition from of knowledge from experts
- o demonstrating of highly complex, real-time, dynamic applications (e.g., Pilot's Assistant)
- o extension of expert systems to domains where human expertise does not exist

Most of the areas or goals of a second generation of expert systems, do not, in general, involve problematic research issues. Thus, the probability that the research efforts will have a significant impact on technology and applications is high. One area where problematic issues may surface is in applying the technology to domains where suitable human expertise does not exist or is not readily accessible, since previous applications have depended on the existence of experts that can introspect about their decision-making.

A "third generation" of expert systems of the type that would be most desirable for advanced intelligent aids is likely to require fundamental advances in knowledge representation and natural language. Such systems would involve integrating reasoning across domains, models of the user (crew) and advanced explanation capabilities. Integrating knowledge sources is quite an advanced question which is just becoming of interest to members of the AI community; there are no well-differentiated schools of thought with respect to it. User models are required to optimize systems response and facilitate interaction with the crew. The appropriate forms for these models is not presently known but their incorporation will undoubtedly require advances in knowledge representation to represent goals, beliefs, intentions, etc. The pilot interviews suggested a strong desire (need?) for more human-like interactions with the intelligent systems. This is not likely to be achieved with direct extensions of the explanation capabilities available in existing systems.

3.2 Knowledge Representation

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 (logic languages, semantic networks and frames), 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

Current work is also considering a new category called hybrid KR systems, in which different components of the system utilize different representation and inference types.

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. The most relevant to the applications of interest here include:

- o representation of actions, events, space, time, mutable objects, beliefs and desires

- representation of users
- support for nonmonotonic reasoning (see Appendix)
- support for analogic reasoning
- support for determining how much time to focus on and reason about a given hypothesis or goal

The representation of actions, events, etc. is critical for the dynamic environment in which aircraft operate. Much that occurs is event-driven and is influenced by external factors. Navigation and planning take place in four-dimensional space-time. The necessity for planning in and reasoning about this environment requires appropriate knowledge representation schemes.

Nonmonotonic reasoning is a critical part of expert system and planning capabilities. The reason is that there is a need to make reasonable assumptions when information is lacking. 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.

Reasoning by analogy is critical to making use of past experience. It includes not only discrimination 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 not analogous. This kind of reasoning can be very relevant for basing current decisions and situation assessment on past, recorded situations.

The need for timely results in the cockpit impose a requirement for speed in arriving at a solution. Hardware advances will help satisfy this requirement, but there may also be a need for explicit control of the computational resources devoted to a problem. 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.

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.

3.3 Planning

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 effect the state of affairs and states do not change without an agent's acting. Of course, there 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:

- o 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 where there are multiple agents, some of whom can be adversaries
- o planning where the environment changes
- o coordination of diverse activities in space and time
- o determining how much effort to expand in trying to find a plan

Many of these gaps relate to those in the knowledge representation and expert systems areas and filling them will require suitable advances in those areas. A few of the more planning-specific issues that will have to be addressed to achieve the kind of advanced capability that is desired are indicated below.

- o Recognition that a plan is no longer valid or is now undesirable due to changing conditions. The environment will not remain static, due to natural events and the action of others; any changes could invalidate a plan.
- o 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.

- Explanation of why a plan has the form it does, rather than some other alternative. Explanation is necessary so that the pilot can know why a particular 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).

- Classification of individual problems as to whether special-purpose (e.g., algorithmic) methods or general-purpose search methods are more appropriate.

3.4 Natural Language

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

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. Today, there are some (pseudo-) natural language systems available commercially, 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 more pragmatic capabilities are not likely to be available until 1988 or 1989. There are no natural language generation products at present.

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, it is highly likely to be successful. Substantial success, sufficient to make systems far more usable than the second generation, is also highly likely because a firm foundation for the research has already been laid and adequate resources should be available as a result of corporate funding and the DARPA Strategic Computing Initiative.

Some of the capabilities required for more advanced natural language systems of the type desirable for cockpit applications are:

- understanding based on a model of user goals and plans (This is necessary for succinct communication without the burden of having to spell out every detail.)

- understanding ill-formed input as the user intended (Input may appear ill-formed due to grammar errors, 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.)

- integration of displays with natural language to convey information (This arises in describing positional information conveyed via map displays, etc.)

- customized natural language generation (Since the language generated in the cockpit may be oriented to many individuals of differing background, expertise, rank, etc., choice of vocabulary and level of description for individuals is important to effective communications. This involves models of user expertise. The purpose is to avoid stating the obvious to different users.)

- natural language understanding and generation in a broad domain (Many different domains arise for "discussion" in the cockpit, e.g., air traffic control, plans, procedures, sensor data, sub-systems.)

With respect to achieving these capabilities, 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, has assumed that the knowledge and reasoning underlying the system is 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 natural language system or an expert system work is a long-term problem requiring programming effort to build or extend these systems.

In short, there is much research required to achieve the potential in natural language processing. 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, modelling 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 applications 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.

3.5 Speech

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, this 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 of the isolated word recognition type. 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.

A recent committee of experts has examined automatic speech recognition in severe environments as follows. (see Appendix). Some of their more relevant conclusions were

- o Current technology for automatic speech recognition is not sufficiently advanced to provide robust, reliable performance in hostile and high-stress environments.

- o Current speech recognition technology is not sufficiently advanced to achieve high performance on continuous spoken input with large vocabularies and/or arbitrary talkers.

- o Current technology is mature enough to support restricted applications in benign environments, with disciplined use under low-stress conditions. Success strongly depends upon the integration of speech recognition with improved automation techniques.

- o No established human-factors methodologies exist for analyzing and evaluating human-machine performance in integrated voice-interactive systems or for systematically quantifying the benefits of speech input as compared to related automation techniques.

- o Speech synthesis is an important adjunct to automatic speech recognition for voice-interactive systems.

Thus, 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.

Based on the expectation of cheaper hardware, special purpose hardware, and incremental improvements in algorithms, we can, nonetheless, confidently project the 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 will become felt as these systems attempt to grow toward high-complexity applications.

To obtain a better understanding of what to expect, consider 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 could be used to assist an aircraft pilot. The vocabulary is small and the task-oriented commands have a rather constrained syntax. The speech will be uttered in a noisy environment, and the speaker could be subject to emotional stress. Such a system must be small enough to be installed in aircraft. A system with this capability should be achievable in 7-8 years given projected support. This assumes the availability of suitable fast computational or special purposes VLSI designs, research on noise handling (signal processing, recognition) and research on speaker stress (speech production, recognition) and 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, that the signal analysis and recognition can be extended to handle the noise, and that the effects of stress can be characterized and handled by known methods.

3.6 AI Tools

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.)

No capabilities in this area are specifically called out by the cockpit application. However, since this area is the infrastructure upon which both AI applications and research are based, it is unquestioningly important. Processing speed is particularly critical for real-time expert systems, planning, and speech recognition. 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.

4. RESEARCH NEEDS IN ARTIFICIAL INTELLIGENCE FOR COCKPIT AIDING

Artificial Intelligence is an extremely diverse and rich field but the following AI sub-areas are particularly important for cockpit aiding: expert systems, knowledge representation, planning, natural language understanding and speech. Important applications such as fault diagnosis draw on aspects of these more fundamental areas. Although progress is being made in each of these areas, the current state of development is not adequate to meet many of the needs associated with the kind of intelligent aiding that seems desirable in light of the analyses in Chapters two and three. Below, we note a few of the special demands that intelligent cockpit aiding will impose on AI development and then separate factors and research needs by AI sub-area; however, because for the most part they are inextricably interrelated when cockpit applications are considered, there will be an inevitable overlapping of topics.

Much of the current interest in applying AI in the cockpit is centered on the potential of Expert Systems (ES) both from the stand-point of narrowly-scoped, "mini-experts", to broader concepts such as a "pilot's associate". It must be recognized that only a few operational systems exist today and none of these are designed to function in as demanding an environment as the cockpit. For example, current rule-based expert systems often break-down when knowledge is inconsistent or incomplete; the behavior exhibited would be unacceptable in a flight-rated system. This raises another important issue with respect to ES (and other AI systems) in the cockpit and that is specification, validation and certification of the programs. This is a critical area in flight applications and it has been virtually ignored in prior AI work. Moreover, it is not at all clear that standard software verification and validation methods, such as they are, are appropriate for these programs.

Perhaps one of the major demands and difficulties for intelligent aids in the cockpit concerns real-time operations. Current AI systems have not operated under the time constraints that the cockpit environment imposes. This is true for expert systems, for planners and for diagnostic systems. Real-time operations could impose severe constraints on the functionality of the aid, the nature of the knowledge representation, the inference mechanisms and the output or display presentation. For example, an intelligent fault monitoring and diagnostic system in the cockpit will have significantly different requirements that one associated with maintaining and repairing equipment in a benign environment. The real-time requirements along with the functional needs will undoubtedly have important implications for the power and architecture of on-board computers, as well as for the development of AI methods.

In general, intelligent cockpit aiding systems will have to "know about" the aircraft and its subsystems, the flight plan, the external world and the crew. Some knowledge will be mathematical quantitative while some will be qualitative. Some information will be numeric and some symbolic. All knowledge and data must be "processed" in appropriate and efficient ways. These requirements will necessitate new developments in knowledge representation and in reasoning techniques.

Fault monitoring and diagnosis is an application for intelligent systems that seems of high value in the cockpit environment. It should be noted that an intelligent system would be one that went considerably beyond the performance and capability of current systems. It would provide more information and/or information about remaining capabilities. It should also be noted that past work in AI on diagnosis has not been focused on the kinds of operational constraints and requirements imposed by the flight deck environment. Thus, to develop intelligent fault monitoring and diagnosis systems will require advances in expert systems, knowledge representation and planning of both a fundamental and targeted nature.

Another area that seems particularly important for intelligent aiding is planning and re-planning. Planning aids can reduce workload. Moreover, adequate representation of plans can improve safety and efficiency by providing a basis for situational awareness. Monitoring the situation in relation to a plan facilitates interpretation of events, discovery of what courses of action are possible and determination of critical situational factors. To achieve such a capability will require significant advances in AI planning.

Intelligent cockpit systems will also have to communicate their often complex results without overloading or confusing the crew and in such a way that the crew maintains control where appropriate. The system will have to integrate qualitative and quantitative information in ways that the crew can interpret quickly. For systems that are viewed as "associates" to the crew, natural language and speech will often be a preferred mode of interface. It will therefore be desirable that the interfaces themselves exhibit some level of intelligence. In particular, they will have to "know enough" about the user to communicate information when it is needed in an appropriate way. In addition, expert systems on board will have to be able to respond to questions so as to provide an adequate basis for the crew to accept their recommendations. Current technology, which allows simple explanation based on simple knowledge structures (rules), is not likely to be adequate in the cockpit. The challenge of developing intelligent interfaces, or interfaces to intelligent systems, or both, is a major aspect of bringing AI to the cockpit.

Finally, we mention a general need for definition of significant test cases and problems appropriate to the transport cockpit. This will ultimately be necessity for measuring the readiness of AI technology and to further identify specific areas requiring additional work. The goal here should be to choose modest sub-problems that have to be solved to solve a larger problem and which illuminate basic difficulties.

The following is a list of some of the most important AI needs that, in our view, will require significant levels of research before the real potential of AI in the cockpit will be realized.

Expert Systems

- Increased domains of expertise - in particular, integration of expertise from various bases such as interpretation of sensor data, symbolic encodings of "situations", flight path management, etc.
- More flexible control structures than simple forward and backward chaining.
- Aircraft and system functional descriptions that are adequate to: infer overall aircraft functionality; to infer causal sequences and other relationships between aircraft systems; to serve as a basis for diagnosis; and to assess the impact of system malfunctions.
- Techniques that can: use default assumptions; reason given incomplete information; and withstand inconsistent data.
- Methods for adding "intelligence" to built-in-test (BIT) systems to reduce false alarm rates.
- Methods for incorporating trends and other information into fault "monitors".
- Methods for facilitating knowledge acquisition.
- Techniques for adapting system output to user intentions and needs.
- Substantial expansion of the explanatory capabilities of expert systems.
- Methods (and hardware) for ensuring real-time operation.
- Methods for specifying, validating and certifying expert system programs.

Knowledge Representation

- Deep conceptual representations for structure, function and process as they relate to aircraft systems and activities. (See above)

- Hybrid knowledge representation systems capable of integrating multiple representations and inference process to provide efficient reasoning in multiple domains.

- Techniques for representing and reasoning with spatial and temporal knowledge.

- Methods for integration both quantitative and qualitative information, including the ability to represent incompletely specified or unknown information.

- Knowledge bases which contain information about graphical presentation and symbology.

- Methods for representing failure (and other) situations.

- Methods for representing (modeling) users.

planning

- Methods for representing flight plans that are computationally adequate for: plan monitoring; high-level pilot interaction for in-flight replanning; assessing importance of deviations from plan; and accounting for the interacting constraints on the flight plan.

- Incremental planners that can interact with a user to refine underspecified plans and revise over-constrained ones.

- Systems that monitor plan execution and revise the plan based on changed situations.

- Planners that can deal with multiple simultaneous and overlapping events both naturally occurring (e.g., weather) and caused by multiple agents.

- Planners that can represent and deal with own intentions and capabilities and those of others.

Natural Language

- Understanding ill-formed input as the user intended (input may be ill-formed because of grammar errors, mispronunciation, faults in the communication medium and/or lack of complete knowledge by the system).
- Integration of natural language with other display modalities or methods (e.g., graphics)
- Customized natural language generation capability.
- Models of user expertise and desires for information so presentation can be tailored to crew (e.g., avoid saying the obvious).

Speech

- Algorithms for recognition of continuous speech that is speaker independent and robust under conditions of degraded input.
- Analysis of pilot/system communication tasks to quantify benefits and potential problems with speech input/output.

5. SUMMARY AND CONCLUDING REMARKS

In this report, we identified areas for, and issues in, the introduction of intelligent aids in a commercial air transport environment. The basic analysis tool was a series of structured interviews with pilots and conducted by interviewers with familiarity with flight operations, human factors and artificial intelligence. Tasks that would benefit from intelligent aids were identified as was the possibility of providing a broader capability in the form of an "intelligent associate" for the crew. Some the key human factors issues pertaining to the introduction of this technology into the cockpit were determined from the interviews and other sources.

The state-of-the-art in Artificial Intelligence areas of prime interest was then reviewed. Here, our approach was to rely heavily on internal expertise for the assessments. Based on this review and the analysis of cockpit tasks, research needs in AI were identified and summarized.

During the interviews, some of which were conducted with the aid of formal questionnaires, four pilots were asked to identify areas of high workload in current operations, to discuss crew performance aids currently in use, and to suggest particular functions and tasks that could, in their judgement, benefit from the introduction and use of intelligent aids. Nine basic flight phases were included in the survey: Pre-Taxi, Taxi Out, Take Off, Climb to Cruise, Cruise, Descend, Approach, Rollout, and Taxi In. Within each of these, nine systems management/monitoring functions were reviewed: Navigation, Electrical, Hydraulic Systems, Flight Control, Fuel, Communications, Thrust, and Internal and External Environment.

The interviews suggested that high levels of workload were associated with Navigation and Communications management and monitoring of the External Environment, and low levels were associated with management of Hydraulic and Electrical systems, when averaged over all flight phases. Descent and Approach phases accounted for the greatest amounts of workload when averaged across all systems management categories. The data also suggested that the design of intelligent aids to crew performance with respect to these systems and phases and, in addition, to Fuel and Thrust management would have high benefit.

A key concept that emerged during the interviews was that of a "Pilot's Associate" which could aid the crew in monitoring and diagnosing faults, establishing priorities among and possibly assist in carrying out courses of action, and generally aid the crew in maintaining an accurate sense of the state of the aircraft and its systems/subsystems. A number of significant human factors issues related to the general question of how, in the context of such an intelligent system, the crew might effectively be kept "in the loop" were identified.

The review of AI suggested that the following areas were of most relevance to cockpit aiding: expert systems, planning, natural language, speech, knowledge representation, and AI tools. The expert system and planning areas are particularly relevant with respect to implementation of specific aids or to a pilot's associate. Natural language and speech are of interest with respect to the crew/system interface. Knowledge representation and AI tools are fundamental aspects of AI that impinge on all developments.

Very briefly, the current and near term state-of-the-art in[4] these areas as they pertain to cockpit-aiding is as follows:

Expert Systems

Expert systems technology is one of the leading "success" stories of AI. Tool kits for building them are commercially available. Despite their successes, they are still severely limited as regards building cockpit aids. They have been applied in only a limited number of domains, to only one domain at once, and typically not in real time. They are usually rule-based, and have the ability to explain what rules were fired in reaching a conclusion. They are weak in handling common sense reasoning, use of default assumptions, and dealing with uncertain, incomplete or inconsistent knowledge.

These weaknesses notwithstanding, expert systems will form a major part of any cockpit aiding approach. We can expect major advances in the next few years as a result of significant private and governmental efforts. Commercially, the representation and control languages should improve in the next few years, and the range of domains attempted will grow. Critical problems such as knowledge acquisition, inadequate computational power and dealing with uncertainty are being addressed by the DARPA Strategic Computing Initiative, and advances in these areas should be significant.

Planning

The ability to predict situations and have ready appropriate responses significantly improves time to response with a viable solution, which is essential for overall effectiveness and

---

[4] Here, we leave AI tools out of the discussion as it is of peripheral interest.

survivability. Current planners are much less mature than expert systems. Planners have been applied in situations where goals and conditions can be well specified, objects and properties are well known and fixed, and actions are physical. They have not been successfully built with multiple agents or to function in real time. Their representation is less mature than that of expert systems also.

Natural Language

Natural language could be one of the best methods of communicating with the pilot. It would allow him to function at his symbolic best, just as though he did have a crew of experts to assist him. Currently there are commercially available pseudo-natural language systems, and a few advanced demonstration systems. They are all restricted to one domain, and don't generally operate in the real-time needed for cockpit application. They don't deal well with ill-formed input. Their vocabularies and grammars are restricting, but will probably support the minimum needed for flight operations.

Speech

Speech is a communication medium which can be significantly less restrictive than switches, touch panels, etc. It is also a medium which pilots find most natural. Combined with natural language, the two form a powerful communication tool which could significantly improve the pilot-vehicle interface. Current speech systems either have small vocabularies (50 or less words) or do not operate in real time, or are speaker dependent using isolated words. The DARPA Strategic Computing Program (SCP) is aiming for a speaker dependent 1000 (connected) word, restricted grammar system. The state-of-the-art for speech recognition in noisy environments is not adequate at this time. However, the SCP is addressing this area as well, and near-term availability of a 200 (isolated) word, speaker independent system is expected.

Knowledge Representation

Different types of aids or a "pilot's associate" will impose needs for representing wide range of different kinds of knowledge, and for successfully drawing inferences across them. The available knowledge representation languages (e.g., LOOPS (LK-TWO) all require programmer expertise, and are weak in representing or supporting inferences about: space and time, actions and events, mutable objects, goals and beliefs and preferences, numeric or qualitative or analogic reasoning. Improvements in general representation will occur in the next few years, but this area will remain the most critical, and, perhaps, least solved of the AI areas.

The most important AI research needs that are specifically related to cockpit aiding as derived from the analysis and review involve: expanding expert systems capability substantially (especially in relation to real-time operation and in dealing with uncertain or incomplete data); developing "intelligent" crew/system interfaces to intelligent systems, including substantial improvement in explanation capability; developing suitable representations for aircraft systems and functionality, particularly to support truly and reliable fault monitoring, diagnosis, and overall status assessment; developing representations of flight plans that are adequate for plan monitoring and to support in-flight replanning; and developing methods for specifying, validating and certifying expert systems programs for flight operations.

We close by noting that, in our view, the full potential of AI in the cockpit can best be realized by an integrated systems approach to the aircraft avionics design aimed at development of an "intelligent airplane". Such an approach is implicit in the notion of pilot's (or crew's) associate. The implementation of that approach will require very careful attention to new design, technological and methodological issues arising because of the possibilities of AI and because of many of its unique characteristics. In the design area critical questions will arise concerning the exact functionality of the system including the nature of the crew system interface. A major feature of an intelligent airplane, we believe, will be a very high-level of "situational awareness" in the system. This can only be achieved by correspondingly high levels of knowledge concerning the aircraft and its sub-systems, the flight plan, etc. and by rapid access to many sources of internal and external data; to achieve this may require fundamental changes in avionics architectures. The technology issues revolve around the ability of AI and computer hardware developments to meet the stringent needs of cockpit aiding. Finally, methodological questions concerning the specification, test and evaluation, documentation and certification for flight-rated AI systems will have to be addressed.

APPENDIX A
ARTIFICIAL INTELLIGENCE: A STATE-OF-THE-ART REVIEW

This appendix provides a state-of-the-art review of six areas of artificial intelligence that were deemed most relevant to the cockpit aiding application: expert systems, knowledge representation, planning, natural language processing, speech, and AI tools. For each of those areas, the following information is provided:

- an overview of the area
- a glossary of key terms and ideas
- the current status of the area including operational applications where they exist and significant problems and research areas
- a brief list of key references

A.1 EXPERT SYSTEMS

A.1.1 Overview

Work in expert systems is currently the most visible area of artificial intelligence. It is also the area most likely to impact the cockpit in the near term. Furthermore, other areas discussed below are closely related to expert system development. For these reasons, we will devote the most consideration to this area of AI.

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 expectation 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."

What is an expert system? 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 a definition 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 that seems too narrow, 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 the system 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 _if_ A & B & C _then_ 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.

We believe that a definition that occupies a mid-ground between the two extreme definitions is the only one that makes sense in the long term. Thus, we consider an _expert system_ to be a computer program that explicitly incorporates knowledge based in significant part on symbolic representation of a body of facts, rules of thumb, strategies, concepts, and heuristics 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 (but not the entirety) 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.

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.

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, a very important step in the process 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.

With the specifications of the intended expert systems in hand, a detailed architecture and representation has to be worked out. 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 A.2 for details regarding Knowledge Representation.)

If the goal is to build an expert system for unsolved problems, then we have a very difficult situation; some of the problems in cockpit aiding may be of this kind in so far as they involve systems or technology for which there are no suitable domain experts. In such a case, the most the expert system designer has to work with are informed guesses by experts in the closest current approximations to the area.

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 A.7). 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.

The relation between the expert system and the user of the system is an important consideration. As is well known, 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, (see section 3.8 for 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.

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, inasmuch as they consisted of a modular collection of rules, they could be extended and modified by simply adding, changing, or deleting rules. It is now recognized 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.

It should be noted that a large number of applications of expert systems pertain to some form of diagnosis. In as much as fault monitoring and diagnosis is a specific area where intelligent aiding is desirable (see Chapter 2) it is worth commenting briefly on work in this area.

There are several techniques relevant to building fault monitoring and diagnostic systems. A monitoring system may be described in terms of "demons". A demon is something that waits for a certain event or set of events to occur and then triggers a set of actions. Thus, it is a means of detecting a fault.

Once a fault has been detected, determination of the most likely cause of the fault is of interest. In the "testing" field this is referred to as fault isolation whereas in AI it is called diagnosis. There are currently two approaches being used in AI diagnostic reasoning:

- Rule based diagnosis.

- Model based diagnosis.

A rule based diagnostic system, such as MYCIN, attempts to reason from a set of failed tests to identify the most likely cause of the failure. It does this by using a set of rules. For example, the failure of a particular test may indicate several potential faults. The set of rules will guide the system to make the appropriate choice, and possibly make additional checks along the way. The knowledge that the system uses to perform the diagnosis is written explicitly in terms of these rules.

An alternative style of diagnosis reasons from a model of the underlying system. This model includes a description of the system in terms of physical and functional structure. The functional structure describes the system in terms of functional units such as and-gates and or-gates. The physical structure of the system is described in such terms as PC-boards or IC-Chips. (See articles by Davis in the Bibliography)

Using this model, a system reasons in much the same way that an engineer would reason about the system given a schematic diagram. Although this approach is likely to be superior to the rule based one, there is much more experience with rule based approaches to diagnosis than the model based ones.

A.1.2 Glossary of Key Words and Ideas

<u>Backward chaining</u>: Reasoning backward from desired conclusions

<u>Causal Reasoning</u>: Reasoning about the causes of observed behavior.

<u>Chaining</u>: Using rules one after the other to draw a complex conclusion in several steps. For instance, if we have two simple rules <u>if</u> A & B <u>then</u> C and <u>if</u> C & D <u>then</u> 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.

<u>Contexting mechanisms</u>: Grouping related rules together, to reduce search.

<u>Diagnosis</u>: Identifying or analyzing the cause or nature of a condition, situation, or problem.

Forward chaining: Reasoning forward from what is initially given.

Hierarchical Models: Multi-level (one level at a time) descriptions of a system so as to minimize the number of parts under considerations any one time.

If-then rule: A pairing of a situation specifications 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.

Monitor: A system which observes the behavior of another system, checking for errors.

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

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

A.1.3 Current Status

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.

To obtain some insight into the current state-of-the-art, we briefly summarize some of the best known expert systems. 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 literature, should be read with the cautions stated 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. (The final report at SRI International appeared in 1978). 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 Society 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.

Despite the above list and the reports in the literature, it is probably the case that, as of September 1984, there were no more than ten fully operational expert system applications in regular use under field conditions. 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, however, 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 given by Gevarter (see reference at the end of section).

Another important aspect of work is expert systems concerns software systems to aid their development. 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.

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 as to the properties of a problem that ensure the development of an effective, operational expert system. We would conjecture that the following features increase the likelihood of successful development.

- o 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.

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

- o An expert system may be <u>decomposable</u>, 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.

- o 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.

- o 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 is easy to model.

The following principal areas are currently the focus of significant investigation in expert systems.

- o 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.

- o 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.

- 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 is 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 feasible to solve, and limited success within five years can be anticipated.

The major gaps in expert systems technology of interest in cockpit applications are listed below. Each of these will be addressed to a major extent over the next few years as a result of the DARPA strategic computing initiative.

- 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 A.2 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 multi-party and adversarial situations. Ignoring disconfirming data, for instance, is not reasonable, since that data may be the key to rejecting a wrong hypothesis.

- "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, multi-processor 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.

In general, it can be said that all known problems in the area of expert systems 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. Of course, new problems and research areas are likely to emerge as more complex and realistic applications are pursued.

A.1.4 Recommended Key References

Aikins, J., Prototypical Knowledge for Expert Systems Artificial Intelligence 20(2):163-210,February,1983.

Boose, J.H., Personal Construct Theory and the Transfer of Human Expertise, Proceedings of AAAI-84, Los Altos, CA: William Kaufmann, Inc., 1984, 27-33.

Buchanan,B. Research on Expert Systems. In Hayes, J., Michie, D., Pao, Y. (editors), Machine Intelligence, pages 269-300. Wiley, New York, 1982.

Buchanan, B.G., Barstow, D., Bechtel, R., Bennett, J., Clancey, W., Kulikowski, C., Mitchell, T., and Waterman, D.A., Constructing an Expert System, in F. Hayes-Roth, D.A. Waterman, and D.B. Lenat, (eds.), Building Expert Systems, Reading, MA: Addison-Wesley Publishing Company, Inc., 1983, 127-168.

Buchanan, B., and Duda, R. Principles of Rule-based Expert Systems.. Technical Report, Stanford University, Department of Computer Sciences,1982.

Campbell, A.,Hollister, V., Duda, R., and Hart, P. Recognition of a Hidden Mineral Deposit by an Artificial Intelligence Program. Science 217:927-929,1982.

Chandrasekaran, B. Expert Systems: Matching Techniques to Tasks. Artificial Intelligence Applications for Business, W. Reitman (ed.), Norwood, NJ: Ablex Publishing Corp., 1984, 41-64.

Davis, R. "Reasoning From First Principles in Electronic Troubleshooting." Int. J. Man-Machine Studies 19 (1983), 403-423.

Davis, R., Shrobe, H., Hamscher, W., Wieckert, K., Shirley, M., Polit, S., Diagnosis Based on Description of Structure and Function. Proc. AAA1-82, 1982, pp. 137-142.

Duda, R., and Reboh, R. AI and Decisions Making: The Prospector Experience, Artificial Intelligence Applications for Business, W. Reitman (ed.), Norwood, NJ: Ablex Publishing Corp., 1984, 111-148.

Duda, R., and Shortliffe, E. Expert Systems Research. Science 220: 260-268, 1983.

Genesereth, M.G., Diagnosis Using Hierarchical Design Models. Proc. AAA1-82, 1982, pp. 278-283

Gevarter, W. An Overview of Expert Systems. National Bureau of Standards Report No. NBSIR 82-2505, October, 1982.

Hayes-Roth, F., Waterman, D., and Lenat, D. (editors). Building Expert Systems. Addison-Wesley, New York, 1983.

Politakis, P., and Weiss, S. Using Empirical Analysis to Refine Expert System Knowledge Bases. Artificial Intelligence V22(1) Jan. 1984, 23-48.

Pople, H. Knowledge-Based Expert Systems: The Buy or Build Decision, Artificial Intelligence Applications for Business, W. Reitman (ed.), Norwood, NJ: Ablex Publishing Corp., 1984, 23-40.

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

Weiss, S., and Kulikowski, C. A Practical Guide to Designing Expert Systems Rowman & Allanheld, New Jersey, 1984.

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

A.2 KNOWLEDGE REPRESENTATION

A.2.1 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 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 databases rather than knowledge representation.

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

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. 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. However, it is all too easy to design a system that infers so much that it is impossible to control, so that while searching for a way to infer a certain fact, it follows many, many blind alleys. Even worse, it is very easy to design systems that cannot guarantee that the questions one might ask are even decidable.

A.2.2 Glossary of Key Words and Ideas

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 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.

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

Subsumption: 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.

A.2.3 Current Status

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 because each of these advances 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.

Styles of description languages for KR systems fall into three 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 is 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 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.

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.

All of the KR approaches discussed above have reached a level of maturity such that the languages have been used in commercial or prototype expert systems or in prototype natural language understanding systems. On the other hand, it is clear that none has yet achieved the level of expressibility or the level of inference support that their creators dream of.

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:

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

The type of reasoning this typifies is called non-monotonic 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.

Representation 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:

- o the need to predict the beliefs and knowledge of colleagues and adversaries in order to appropriately assess, plan, etc.
- o the need of some expert systems to reason about likely adversarial action

- o 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)
- o 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.

All of the problem areas discussed are being pursued.

A.2.4 Recommended Key References

Bobrow, D.G., (ed.) Artificial Intelligence, Special Issue on Non-Monotonic Logic, vol. 13, no. 1-2, April, 1980

Bobrow, D.G. and Collins, A. (eds) Representation and Understanding New York; Academic Press, Inc., 1975

Bobrow, D.G. and Winograd, T., An Overview of KRL, a Knowledge Representation Language, Cognitive Science, vol. 1, no. 1, January, 1977, pp. 3-46

Brachman, R.J., Fikes, R.E., and levesque, H.J., KRYPTON: A Functional Approach to Knowledge Representation, IEEE Computer Special Issue on Knowledge Representation, Oct., 1983, pp. 67-73.

Brachman, R.J. and Levesque, H.J., The Tractability of Subsumption in Frame-Based Description Language, Proc. of AAAI-84, Los Altos, CA: William Kaufmann, Inc., 1983.

Brachman, R.J. and Schmolze, J.G., An Overview of the KL-ONE Knowledge Representation System, Cognitive Science, vol. 9, no. 2, Apr-June 1985.

Brodie, L., Mylopoulos, J. and Schmidt, J.W. (eds.) On Conceptual Modeling, New York: Springer-Verlag, 1984. pp. 348-351.

Carbonell, J., Derivational Analogy and its Role in Problem-Solving Proceedings of AAAI-83 Los Altos, CA: William J. Kaufmann, Inc. 1983. pp.64-69.

Findler, V. (ed.), Associative Networks, NY: Academic Press, 1979.

Hayes, P.J., In Defense of Logic Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Pittsburgh, PA: Dept. of Computer Science, Carnegie-Mellon University 1977, pp.559-565

Hayes, P.J., The Logic of Frames in Frame Conceptions and Text Understanding, Dieter Metzing (ed.). Berlin: Walterde Gruyter, 1980. pp.46-61.

Levesque, H.J., A Fundamental Tradeoff in Knowledge Representation and Reasoning, Proc. of CSCSI-84, London, Ontario, May, 1984, pp.141-152.

Levesque, H.J., A Logic of Implicit and Explicit Belief Proc. of AAAI-83 Los Altos, CA: William Kaufmann, Inc., 1983.

Rich, E., Default Reasoning as likelihood Reasoning Proceedings of AAAI-83 Los Altos, CA: William J. Kaufmann, Inc., 1983.

Warren, D.H.D., Generating Conditional Plans and Programs, Proceedings of the AISB Summer Conference, Edinburgh, 1976, 344-354.

Woods, W.A., What's in a Link: Foundations for Semantic Networks Representation and Understanding, D.G. Bobrow and A. Collins (eds.), New York: Academic Press, Inc., 1975, pp.35-82.

A.3 PLANNING

A.3.1 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 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, it is necessary to make explicit any time restrictions that are imposed. 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 that can be allocated, the restrictions on them, etc. Likewise, one must 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 so specified.

Once such planning policy questions are recognized and made explicit, it becomes apparent that classical research on planning has been carried out under highly unrealistic assumptions with respect to real world planning conditions.

An 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 pre-conditions 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.

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.

A.3.2 Glossary of Keywords and Ideas

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 goal is acceptable and beyond merely achieving the goal. Examples include the cost and the 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 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.

A.3.3 Current Status

Broadly speaking, the conceptual development of the planning 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 meta planning, 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.

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) and in terms of many successive rule applications that may be needed to infer a conclusion.

Current planners cannot solve the planning problems that arise in a complex domain involving both time and external events. The principal reason for this is that the knowledge representation systems that underlie them cannot be used to express many of the problems that are part of planning. As we have discussed earlier, classical planning research has for the most part focused on 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 many domains of interest, 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 the objective may become more constrained, sometimes to the point of being over constrained. 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.

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 may not be under the planner's total control, with nature and other parties affecting the situation. 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 actions of others may to thwart or these plans or cause them to be modified.

All of the current principal areas of research in planning share the framework described earlier of search through a state space, though the notion of "node" and "arc" may differ widely. These research areas include the following.

- 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 Robinson 81], [Vere 81], [Wesson & Hayes-Roth 79], [Sacerdoti 74].

- 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 [Pearl 1983] and [Hayes-Roth and Hayes-Roth, 1978].

- 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 micro-processors. See [Konolige & Nilsson 80], [Thorndyke et al 81] for initial work in this case.

- 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 [Thorndyke et al 81]. [Wesson & Hayes-Roth 79], [Wilkins & Robinson 81], [Srinivas 78].

- 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 [Wilkins & Robinson 81] and [Pearl 83].

Since planning has generally been simplified by considering only a single agent and an unchanging situation, an obvious gap in current systems particularly as they relate to cockpit aids, is planning in dynamic environments. In such an environment, it is possible that the situation may change, either due to natural causes or to the activities of other agents operating in the environment.

In these dynamic 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:

- determine the significant, relatively invariant features of the environment

- understand what their implications are

- provide early alerts to significant changes in the environment that might effect current plans

Another gap involves developing plans in situations involving communication among multiple actors. Planning research aimed at this problem could be of significant practical importance in a military context and in air traffic situations.

Still another gap in current work 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 is needed to develop:

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

- 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

A.3.4 Recommended Key References

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

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

Hayes-Roth, B., Hayes-Roth, F., Shapiro, N. & Wescourt,

K. _Planner's workbench: A computer aid to replanning_ Technical Report, _The Rand Corporation_, Santa Monica, CA, 1981

Konolige, K. & Nilsson, N.J. Multiple-agent planning systems. In _Proceedings of the First Annual National Conference on Artificial Intelligence_, pages 138-142. AAAI, Stanford U., CA, August 1980.

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, and E.A.Stohr (editors), _Decision Support Systems_, North Holland, Amsterdam, 1982.

Rich, Charles. A Formal Representation for Plans in the Programmer's Apprentice. In _Proceedings of the 7th Int'l. Joint Conference on Artificial Intelligence_, pages 1044-1052. IJCAI, August 1981.

Sacerdoti, E.D. _A Structure for Plans and Behavior_. Elsevier North-Holland, NY, 1977. (Artificial Intelligence Series).

Sridharan, N.S. and Bresina, J.L. Plan formation in large, realistic domains. In Fourth National Conference of CSCSI/SCEIO, pages 12-18. Canadian Society for Computational Studies of Intelligence/Society Canadienne pour Etudes d'intelligence par Ordinateur, May, 1982. (Available as Rutgers Tech. Rept. RU-CBM-TR-1)

Stefik, M. _Planning with constraints_. PhD thesis, Stanford University, January, 1980. (Report STAN-CS-80-784).

A.4 NATURAL LANGUAGE

A.4.1 Overview

The goal of work in this area is to enable computers to communicate in natural language. By this it is meant 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 person 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 several reasons why natural language understanding by a computer system is desirable:

- some useful input may not be available in any other form, such as newspaper articles or the comments field of even what is otherwise a very stylized, constrained message.

- 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.

- 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.

- even frequent users have facilities which they use infrequently and therefore for which natural language will prove convenient.

- 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.

- 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:

- 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

- 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

- 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 of the cryptic dialogue with the 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:

- 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.

- 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?"

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

- 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.

- 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.

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

- 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 they want, 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.

A.4.2 Glossary of Key Words and Ideas

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) of 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 <u>illocutionary force</u> (e.g., asking a question, making a statement, making a promise, etc.)

2. Its <u>propositional content</u> (the description of what is asked, stated, promised, etc.)

<u>Syntax</u>: 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.

A.4.3 Current Status

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 database 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 database: "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 database system sometime 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.

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)

- o a tightly scoped and restricted domain (e.g., a particular data base)

- o a knowledge base for this domain (e.g., the facts)

- o inference methods

- o 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.

There have been other approaches to building natural language understanding systems -- syntax-free semantics (Schank); semantics-free syntax; keyword analysis; and various kinds of mathematically-based models, (e.g, Markov models of natural language). Few researchers believe these approaches to be adequate (a notable exception being Shank and some of his students). 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 a 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 as 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 database environment, the only way to distinguish between the following two requests is by syntax (which conveys the intended message).

1. "List all assets of any company that were sold to XYZ in 1984."

2. "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 or purely stylistic feedback to authors editing their manuscripts.

One major focus of current research 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 reference 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 <u>produce</u> 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. The major bottleneck is the time and effort involved in modeling increasingly broad and complex subject domains.

If the current research outlined above is successful, it is likely that a system that can understand substantial amounts of human conversation will be produced. It would function as a very literal-minded, narrow, but nonetheless, quite useful assistant that can communicate with us.

Besides the areas listed above, there are three additional important 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 for future development 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.

A.4.4 Recommended Key References

Allen, J. (ed.) Special Issue on Ill-Formed Input, American Journal of Computational Linguistics. 9, 3-4, 1983.

Bates, M. and 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. and Berwick, R.C. (eds), Computational Models of Discourse, Cambridge, MA: The MIT Press, 1983.

Cerone, N., (ed.), Computational Linguistics, Oxford: Pergamon Press, 1983.

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

Johnson, C. and Bachenko, J., Applied Computational Linguistics in Perspective: Proceedings of the Workshop, American Journal of Computational Linguistics, 8, 2, 1982, 55-83.

Joshi, A., Webber, B., and 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. and Riesbeck, C.K., (eds.), Inside Computer Understanding, Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers, 1981.

Sparck-Jones, K. and 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.

A.5 SPEECH

A.5.1 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.

A.5.2 Glossary of Keywords and Ideas

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. "did you"), 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.

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

<u>Speed of Operation</u>: 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.

A.5.3 Current Status

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 is itself 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.

The state-of-the-art of automatic speech recognition is severe environments was addressed in a very recent study by a committee of the National Research Council. The report of that Committee (Anon, 1984) listed both conclusions and recommendations with respect to the area that represent the last current thinking on the subject of thinking of experts. These are included here, verbatim, for ease of reference.

Based on its exposure to the issues and its familiarity with the field, the committee concluded that:

- The use of speech for communication between humans and machines has distinct potential for military and other government purposes.

- Current technology for automatic speech recognition is not sufficiently advanced to provide robust, reliable performance in hostile and high-stress environments.

- Current speech recognition technology is not sufficiently advanced to achieve high performance on continuous spoken input with large vocabularies and/or arbitrary talkers.

- Current technology is mature enough to support restricted applications in benign environments, with disciplined use under low-stress conditions. Success strongly depends upon the integration of speech recognition with improved automation techniques.

- No standardized techniques exist for evaluation and comparing the performance of speech recognizers.

- No established human-factors methodologies exist for analyzing and evaluation human-machine performance in integrated voice-interactive systems or for systematically quantifying the benefits of speech input as compared to related automation techniques.

- There is insufficient fundamental understanding of how human speech degrades under severe environmental and stress conditions and of how to design recognition algorithms for these conditions.

- Government-sponsored efforts are currently insufficient to sustain major advances in speech recognition technology.

- Laboratory studies of speech recognition algorithms will probably require sophisticated computational resources that are not widely available.

- Successful deployment of advanced speech recognition systems will be directly related to, and in part dependent upon, continued advances in integrated circuit technology and computer architecture.

- Speech synthesis is an important adjunct to automatic speech recognition for voice-interactive systems.

- No central focus exists in the U.S. government to manage research and development in speech recognition.

The committee's conclusions lead to corollary recommendations. These recommendations aim to achieve a speech recognition technology that can provide utility, accuracy, and reliability in severe as well as benign environments.

- A basic research program is needed to characterize speech and its variabilities, including the study of the acoustic properties of speech in various contexts and for different speakers.

- Because an automatic speech recognizer is limited by the information delivered to it, new methods for sound transduction (including microphone systems designed for severe environments) and for electronic signal enhancement should be sought and studied.

- Significant research efforts are required in the design of algorithms and systems for the recognition of continuous speech in complex application domains, for speaker-independent operation, and for robust performance under conditions of degraded input.

- Research is necessary to establish human-factors procedures for analyzing human-machine communication tasks, to quantify the benefits that speech input/output can contribute, and to develop systematic techniques for integrating speech functions into the systems design.

- Extensive hardware development and deployment, based on existing technology, is inappropriate. Exploratory hardware efforts, however, are vital for gaining practical knowledge about applications and for establishing the limitations of existing technology.

- Standardization should be established to quantify the performance of automatic speech recognizers and to permit comparisons among algorithm philosophies and environments. Common data bases and prescribed procedures for assessing performance should be made generally available.

- Sophisticated computational capabilities are required to support continued advances in speech recognition work. A program for advanced research in speech recognition should have appropriate interfaces with government-sponsored work on high-speed processors and strategic computing.

- A substantial, sustained, and coordinated program of research and development is required to realize the potential of speech recognition within the U.S. government. The program should be built around long-range goals, with the acquisition of fundamental knowledge as a central thrust. This objective is especially crucial to advancing continuous speech recognition and to achieving talker independence with large vocabularies. A focus of responsibility and accountability as well as a means for coordinating the program is necessary.

A.5.4 Recommended Key References

Committee on Computerized Speech Recognition, National Research Council, Automatic Speech Recognition in Severe Environments, Washington, DC: National Academy Press, 1984.

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" (North-Holland Publishing Co., Amsterdam 1973).

Walker, D.E. "Speech Understanding, Computational Linguistics, and Artificial Intelligence", Stanford Research Institute, Menlo Park, CA, Artificial Intelligence Center Note 85 (1973).

Wolf, J.J. "Speech Recognition and Understanding", in K.-S. Fu (ed.), Digital Pattern Recognition, 2nd ed. (Springer-Verlag, Berlin, 1980).

Lea, W.A., "Selecting the Best Speech Recognizer for the Job", Speech Technology, Vol. 1, Jan/Feb 1983, pp. 10-29.

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

Simon, J.C. (ed.), Spoken Language Generation and Understanding, (Reidel, Dordrecht, 1980).

Haton, J.P. (ed.), "Automatic Speech Analysis and Recognition" (Reidel, Dordrecht, 1982).

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

DeMori, R. and Laface, P. "Use of Fuzzy Algorithms for Phonetic and Phonemic Labeling of Continuous Speech", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, pp. 136-148, March 1980.

Bahl, L.R., Jelinek, F., and Mercer, R.L. "A Maximum Likelihood Approach to Continuous Speech Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, pp. 179-190, March 1983.

A.6 AI Tools and Environments

A.6.1 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 of AI research. 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 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.

A.6.2 Glossary of Keywords and Ideas

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.

A.6.3 Current Status

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.

A major factor governing effectiveness of programming tools 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.

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. However, though tools can lighten the burden of the software effort in constructing AI systems will be a burden for the foreseeable future.

An ongoing concern is improvements in cost and performance of systems for AI programming, but advances in VLSI will continue to offer substantial improvements in both.

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.

A.6.4 Recommended Key References

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

Goldberg, A., and Robson, D. Smalltalk-80: The language and its implementation, Reading, MA: Addison-Wesley, 1983

Sanella, M. (ed.), Interlisp Reference Manual, Xerox Corp., 1983

Sheil, B., Power Tools for Programmers, Datamation, Feb. 83

Stefik, M., Bobrow, D., Mittal, S., Conway, L., Knowledge Programming in LOOPS, AI Magazine, 1983.

Weinreb, D. and Moon D., Lisp Machine Manual, Symbolics, Inc., 1981.

1. Report No. NASA CR-3944	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ANALYSIS OF THE APPLICATION OF AI TO THE DEVELOPMENT OF INTELLIGENT AIDS FOR FLIGHT CREW TASKS		5. Report Date October 1985
		6. Performing Organization Code 5937
7. Author(s) Sheldon Baron and Carl Feehrer		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Bolt Beranek and Newman Inc. 10 Moulton Street Cambridge, MA 02238		11. Contract or Grant No. NAS1-17335
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code 505-35-13-04

15. Supplementary Notes

Langley Technical Monitor: Kathy H. Abbott

16. Abstract

This report presents the results of a study aimed at developing a basis for applying AI to the flight deck environment of commercial transport aircraft. In particular, the study was comprised of four tasks: (1) Analysis of flight crew tasks, (2) Survey of the state-of-the-art of relevant AI areas, (3) Identification of human factors issues relevant to intelligent cockpit aids, and (4) Identification of AI areas requiring further research.

17. Key Words (Suggested by Author(s)) artificial intelligence, cockpit aids, human factors, civil transport, natural language, expert systems, speech io, planning, knowledge representation	18. Distribution Statement Unclassified - Unlimited Subject Category 03

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 116	22. Price A06

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1985

End of Document