LOAN DOCUMENT

PHOTOGRAPH THIS SHEET

LEVEL

0

INVENTORY

DTIC ACCESSION NUMBER

TM-85838

DOCUMENT IDENTIFICATION

Oct 83

H A N D L E   W I T H   C A R E

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I ☑
DTIC TRAC ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/
AVAILABILITY CODES

DISTRIBUTION	AVAILABILITY AND/OR SPECIAL	
A-1		

DISTRIBUTION STAMP

DATE ACCESSIONED

DATE RETURNED

19990519 132

DATE RECEIVED IN DTIC

REGISTERED OR CERTIFIED NUMBER

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-FDAC

DTIC FORM JUN 90 70A

DOCUMENT PROCESSING SHEET

PREVIOUS EDITIONS MAY BE USED UNTIL STOCK IS EXHAUSTED.

LOAN DOCUMENT

Technical Memorandum 85838

An Overview of Artificial Intelligence and Robotics

VOLUME I — ARTIFICIAL INTELLIGENCE
Part B — Applications

William B. Gevarter

OCTOBER 1983

25th Anniversary
1958-1983

NASA

NASA Technical Memorandum 85838

An Overview of Artificial Intelligence and Robotics

VOLUME I — ARTIFICIAL INTELLIGENCE
Part B — Applications

William B. Gevarter
OFFICE OF AERONAUTICS AND SPACE TECHNOLOGY
NASA Headquarters
Washington, D.C. 20546

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1983

PREFACE

The real payoff for artificial intelligence (AI) is applications. It is applications that has thrust AI into prominence and commercialization in the 1980's. This report presents overviews of key application areas: Expert Systems, Computer Vision, Natural Language Processing, Speech Interfaces, and Problem Solving and Planning. The basic approaches to such systems, the state of the art, existing systems and future trends and expectations are covered.

It is anticipated that this report will prove useful to engineering and research managers, potential users and others who will be affected by the rapidly growing area of AI applications.

This report is part of the NBS/NASA series of overviews on AI and Robotics. Due to the scope of AI, Volume I — Artificial Intelligence — is issued in three parts (this report being Part B):

Part A: The Core Ingredients, NASA TM 85836, June 1983

- **I.** Artificial Intelligence—What It Is
- **II.** The Rise, Fall and Rebirth of AI
- **III.** Basic Elements of AI
- **IV.** Applications
- **V.** The Principal Participants
- **VI.** State-of-the-Art
- **VII.** Towards the Future

Sources for Further Information

Glossary

Part B: Applications, NASA TM 85838, Sept. 1983

- **I.** Expert Systems
- **II.** Computer Vision
- **III.** Natural Language Processing
- **IV.** Speech Recognition and Speech Understanding
- **V.** Speech Synthesis
- **VI.** Problem-Solving and Planning

Part C: Basic AI Topics, NASA TM 85839, Oct. 1983

- **I.** Artificial Intelligence and Automation
- **II.** Search-Oriented Automated Problem Solving and Planning
- **III.** Knowledge Representation
- **IV.** Computational Logic

ACKNOWLEDGMENTS

I wish to thank the many people and organizations who have contributed to this report, both in providing information, and in reviewing the report and suggesting corrections, modifications and additions. I particulary would like to thank Terry Cronin of U.S. Army Signal Warfare Lab., Dave Pallett and Jim Hieronymus of NBS, Ed Zenker of NASA Goddard Space Flight Center, Dave Gilblom, Carl Berney and Gabe Groner of Speech Plus, Jude Franklin of the U.S. Navy Center for Applied Research in AI, and David R. Brown of SRI International for their review of various portions of this report and their many helpful suggestions. However, the responsibility of any remaining errors or inaccuracies must remain with the author.

It is not the intent of NASA or the National Bureau of Standards to recommend or endorse any of the products, manufacturers or organizations named in this report, but simply to attempt to provide an overview of the AI field. However, in a diverse and rapidly changing field such as AI, important activities, organizations and products may not have been mentioned. Lack of such mention does not in any way imply that they are not also worthwhile. The author would appreciate having any such omissions or oversights and any residual errors or inaccuracies called to his attention so that they can be considered for future reports.

TABLE OF CONTENTS

Page

Preface iii
Acknowledgments iv
Foreword xi

I. Expert Systems 1
- A. Introduction 1
- B. What is an Expert System? 1
- C. The Basic Structure of an Expert System 1
- D. The Knowledge Base 3
- E. The Control Structure 3
- F. Uses of Expert Systems 3
- G. Architecture of Expert Systems 3
- H. Existing Expert Systems 8
- I. Constructing an Expert System 8
- J. Summary of the State-of-the-Art 13
- K. Future Trends 14
- References 16

II. Computer Vision 17
- A. Introduction 17
- B. Definition 17
- C. Relation to Human Vision 17
- D. Basis for a General Purpose Image Understanding System 19
- E. Basic Paradigms for Computer Vision 21
  1. Hierarchical Bottom-up Approach 21
  2. Hierarchical Top-down Approach 21
  3. Heterarchical Approach 21
  4. Blackboard Approach 23
- F. Level of Representation 23
- G. Research in Model-Based Vision Systems 24
- H. Industrial Vision Systems 28
  1. General Characteristics 28
  2. Examples of Industrial Visual Inspection Systems 28
  3. Examples of Efforts in Industrial Visual Recognition and Location Systems . . . 29
  4. Commercially Available Industrial Vision Systems 29
- I. Who Is Doing It? 29
  1. Research Oriented 29
  2. Commercial Vision Systems Developers 34

TABLE OF CONTENTS (continued)

Page

J. Summary of the State-of-the-Art 34
  1. Human Vision 34
  2. Low and Intermediate Levels of Processing 34
  3. Industrial Vision Systems 35
  4. General Purpose Vision Systems 35
K. Applications and Future Trends 36
  1. Techniques 36
  2. Hardware and Architecture 36
  3. AI and General Vision Systems 36
  4. Modelling and Programming 38
  5. Knowledge Acquisition 38
  6. Sensing 38
  7. Industrial Vision Systems 38
  8. Future Applications 39
L. Conclusion 39
References 39

III. Natural Language Processing 41
A. Introduction 41
B. Applications 41
C. Approach 42
  1. Type A: No World Models 42
  2. Type B: Systems that Use Explicit World Models 43
  3. Type C: Systems that Include Information About the Goals and Beliefs of Intelligent Entities 43
D. The Parsing Problem 43
E. Grammars 43
  1. Phrase Structure Grammar - Context Free Grammar 43
  2. Transformational Grammar 44
  3. Case Grammar 44
  4. Semantic Grammar 44
  5. Other Grammars 44
F. Semantics and the Cantankerous Aspects of Language 45
  1. Multiple Word Senses 45
  2. Pronouns 45
  3. Ellipsis and Substitution 45
G. Knowledge Representation 45
  1. Procedural Representations 45
  2. Declarative Representations 46
  3. Case Frames 46

TABLE OF CONTENTS (continued)

Page

4. Conceptual Dependency 46
5. Frames 46
6. Scripts 46

H. Syntactic Parsing 46
1. Template Matching 46
2. Transition Nets 46
3. Other Parsers 49

I. Semantics, Parsing and Understanding 49

J. Natural Language Processing (NLP) Systems 49
1. Kinds 49
2. Research NLP Systems 50
3. Commercial Systems 50

K. State of the Art 52

L. Principal U.S. Participants in NLP 52
1. Research and Development 52
2. Principal U.S. Government Agencies Funding NLP Research 54
3. Commercial NLP Systems 54
4. Non-U.S. 54

M. Forecast 54

References 55

IV. Speech Recognition and Speech Understanding 57

A. Introduction 57

B. Applications 57

C. The Nature of Speech Sounds 57

D. Isolated Word Recognition 59

E. Recognizing Continuous Speech 61

F. Speech Understanding 61

G. The ARPA Speech Understanding Research (SUR) Project 63
1. Introduction 63
2. HEARSAY II 63
3. HARPY 65
4. HWIM 65
5. Summary of the ARPA SUR Program 65

H. State of the Art 69
1. Speech Recognition 69
2. Speech Understanding 69

I. Who Is Doing Speech Recognition Related Work 69
1. Commercial Organizations 69
2. Universities 71

TABLE OF CONTENTS (continued)

	Page
J. Problems and Issues	71
K. Future Trends	72
References	72
V. Speech Synthesis	74
A. Introduction	74
B. Why Synthesis	74
C. Human Speech	74
D. Electronic Simulation of the Speech Mechanism	76
E. Synthesis in Speech Compression and Regeneration	76
F. Parametric Coding Schemes	
1. Introduction	78
2. Formant Coding	78
3. Linear Predictive Coding	78
4. PARCOR	78
5. Line Spectrum Pair	78
6. Parametric Waveform Coding	79
G. Waveform Coding Schemes	79
1. ADPCM	79
2. Mozers Waveform Coding	79
H. Coding the Words To Be Stored	79
I. Generating Speech for Text	80
J. State of the Art	81
K. Some Available Commercial Systems	83
L. Problems and Issues	83
M. Forecast	83
References	86
VI. Problem Solving and Planning	87
A. Introduction	87
B. Planning Defined	87
C. Basic Planning Paradigm	87
D. Paradigms for Generating Plans	90
1. Non-Hierarchical Planning	90
2. Hierarchical Planning	90
3. Utilization of Skeletal Plans	92
4. Opportunistic Planning	92
E. Planners	92
F. Trends	92
References	108

TABLES

		Page
I-1.	Characteristics of Example Expert Systems	4
	- 1 DENDRAL	4
	- 2 AM	5
	- 3 R1	6
	- 4 MYCIN	7
I-2.	Control Structures of Some Well Known Expert Systems	9
I-3.	Existing Expert Systems by Function	10
II-1.	Model-Based Vision Systems	25
	- 1 Acronym	25
	- 2 Visions	27
II-2.	Example Research Efforts in Industrial Visual Recognition and Location Systems	30
III-1.	Some Applications of Natural Language Processing	42
III-2.	Some Commercial Natural Language Systems	51
IV-1.	Speech Recognition Applications	58
IV-2.	Speech Understanding Applications	59
IV-3.	Summary of ARPA's Speech Understanding Systems	67
IV-4.	TI's Test of Speech Recognizers on Individual Words	70
V-1.	Applications of Speech Synthesis	75
V-2.	Available Commercial Synthesizer Systems	84
VI-1.	Planners	94
	- 1 STRIPS	94
	- 2 HACKER	95
	- 3 INTERPLAN	96
	- 4 Waldinger's	97
	- 5 ABSTRIPS	98
	- 6 NOAH	99
	- 7 MOLGEN I	100
	- 8 MOLGEN II	102
	- 9 DEVISER	103
	-10 KNOBS	105
	-11 ISIS II	106

FIGURES

		Page
I-1.	Basic Structure of an Expert System	2
I-2.	Expert Systems Applications Now Under Development	14
I-3.	Future Opportunities for Expert Systems	15
II-1.	A Framework for Early and Intermediate States in A Theory of Visual Information Processing	18
II-2.	An Example of a 2 ½ D Sketch	19
II-3.	Model-based Interpretation of Images	20
II-4.	Basic Image Understanding Paradigms	22
II-5.	Organization of a Visual System	23
II-6.	Examples of Applications of Computer Vision Now Underway	37
III-1.	A Transition Network for a Small Subset of English	47
III-2.	Example Noun Phrase Decomposition	48
III-3.	Parse Tree Representation of the Noun Phrase Surface Structure	48
IV-1.	Basic Speech Recognition Paradigm	60
IV-2.	The Processing Hierarchy in Speech Understanding	62
IV-3.	A Block Diagram of the CMU HEARSAY-II System Organization	64
IV-4.	Block Diagram of the BBN HWIM Organization	66
V-1.	A Simplified Diagram of the Computalker CT-1 Parametric Synthesizer	77
V-2.	Recording and Reproduction of Speech Using a Compressed Speech System	77
V-3.	Text to Speech Synthesis	80
V-4.	Speech Quality Versus Bit Rate for Various Coding Schemes	81
V-5.	Text to Speech Conversion	82
VI-1.	Wilensky Planning Paradigm	88
VI-2.	Nature of Goal Conflicts	89
VI-3.	Resolving Goal Conflicts by Replanning	91
VI-4.	Planning Techniques	93
VI-5.	Simplified Flow Chart of DEVISER's Core Planner	107

FOREWORD

The opening of the decade of the 80's saw Artificial Intelligence (AI) transition from a primarily research topic to commercial applications. The full impact of this transition has yet to be felt.

AI has been designated by the U.S. Defense Science Board as one of the top 10 major payoff areas for the military. It has been made the core ingredient of Japan's Fifth Generation computer research project by which they seek to catapult Japan into the dominant information society in the 1990's. Similar importance has been attached to AI in the U.S., Great Britain and France.

This report summarizes the key AI application areas of Expert Systems, Computer Vision, Natural Language Processing, Speech Interfaces, and Problem Solving and Planning. More detailed information can be found in the following documents available from the National Technical Information Service (NTIS), Springfield, VA 22161.

An Overview of Expert Systems, NBSIR 2505
May 1982 (Revised October 1982)

An Overview of Computer Vision, NBSIR 2582
September 1982

An Overview of Natural Language Processing
NBSIR 83-2687, April 1983
NASA TM 85635, April 1983

Two emerging AI topics — Automatic Programming, and Machine Learning — are not treated separately in this report but are included under Expert Systems.

This document is Part B of the three part report:

An Overview of Artificial Intelligence and Robotics
Volume I — Artificial Intelligence

Part A — The Core Ingredients, NASA TM 85836, June 1983
Part B — Applications, NASA TM 85838, Sept. 1983
Part C — Basic AI Topics, NASA TM 85839, Oct. 1983

The important AI application areas of robotics and automated manufacturing are treated in *An Overview of Artificial Intelligence and Robotics*
Volume II — Robotics, NBSIR 82-2479, March 1982.

I. EXPERT SYSTEMS

A. Introduction

Expert Systems is probably the "hottest" topic in Artificial Intelligence (AI) today. Prior to the last decade, in trying to find solutions to problems, AI researchers tended to rely on non-knowledge-guided search techniques or computational logic. These techniques were successfully used to solve elementary problems or very well structured problems such as games. However, real complex problems are prone to have the characteristics that their search space tends to expand exponentially with the number of parameters involved. For such problems, these older techniques have generally proved to be inadequate and a new approach was needed. This new approach emphasized knowledge rather than search and has led to the field of Knowledge Engineering and Expert Systems. The resultant expert systems technology, limited to academic laboratories in the 70's, is now becoming cost-effective and is beginning to enter into commercial applications.

B. What is an Expert System?

Feigenbaum, a pioneer in expert systems, (1982, p.1) states:

> An "expert system" is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field.
>
> The knowledge of an expert system consists of facts and heuristics. The "facts" constitute a body of information that is widely shared, publicly available, and generally agreed upon by experts in a field. The "heuristics" are mostly private, little-discussed rules of good judgement (rules of plausible reasoning, rules of good guessing) that characterize expert-level decision making in the field. The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses.

It has become fashionable today to characterize any large, complex AI system that uses large bodies of domain knowledge as an expert system. Thus, nearly all AI applications to real-world problems can be considered in this category, though the designation "knowledge-based systems" is more appropriate.

C. The Basic Structure of an Expert System

An expert system consists of:

(1) a knowledge base (or knowledge source) of domain facts and heuristics associated with the problem;

(2) an inference procedure (or control structure) for utilizing the knowledge base in the solution of the problem;

(3) a working memory — "global data base" — for keeping track of the problem status, the input data for the particular problem, and the relevant history of what has thus far been done.

A human "domain expert" usually collaborates to help develop the knowledge base. Once the system has been developed, in addition to solving problems, it can also be used to help instruct others in developing their own expertise.

It is desirable, though not yet common, to have a user-friendly natural language interface to facilitate the use of the system in all three modes: development, problem solving, instruction. In some sophisticated systems, an explanation module is also included, allowing the user to challenge and examine the reasoning process underlying the system's answers. Figure I-1 is a diagram of an idealized expert system. When the domain knowledge is stored as production rules, the knowledge base is often referred to as the "rule base," and the inference engine as the "rule interpreter."

An expert system differs from more conventional computer programs in several important respects. Duda (1981, p. 242) observes that, in an expert system ". . . there is a clear separation of general knowledge about the problem (the rules forming a knowledge base) from information about the current problem (the input data) and the methods for applying the general knowledge to the problem (the rule interpreter)." In a conventional computer program, knowledge pertinent to the problem and methods for utilizing this knowledge are all intermixed, so that it is difficult to change the program. In an expert system, ". . . the program itself is only an interpreter (or

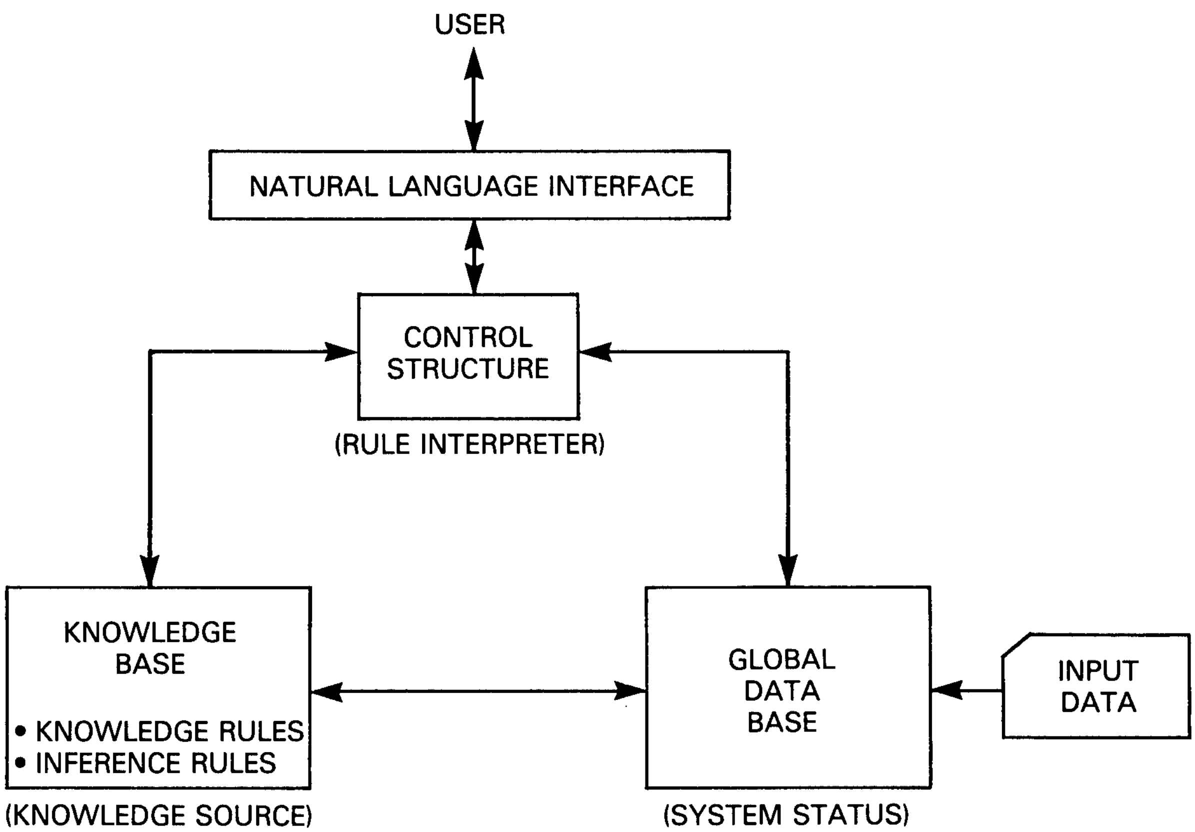


Figure I-1. Basic Structure of an Expert System.

general reasoning mechanism) and (ideally) the system can be changed by simply adding or subtracting rules in the knowledge base.''

D. The Knowledge Base

The most popular approach to representing the domain knowledge (both facts and heuristics) needed for an expert system is by production rules (also referred to as ''SITUATION-ACTION rules'' or ''IF-THEN rules'').* Thus, often a knowledge base is made up mostly of rules which are invoked by pattern matching with features of the task environment as they currently appear in the global data base.

E. The Control Structure

In an expert system a problem-solving paradigm must be chosen to organize and control the steps taken to solve the problem. A common, but powerful approach involves the chaining of IF-THEN rules to form a line of reasoning. The rules are actuated by patterns (which, depending on the strategy, match either the IF or the THEN side of the rules) in the global data base. The application of the rule changes the system status and therefore the data base, enabling some rules and disabling others. The rule interpreter uses a control strategy for finding the enabled rules and for deciding which of the enabled rules to apply. The basic control strategies used may be top-down (goal driven), bottom-up (data driven), or a combination of the two that uses a relaxation-like convergence process to join these opposite lines of reasoning together at some intermediate point to yield a problem solution. However, virtually all the heuristic search and problem solving techniques that the AI community has devised have appeared in the various expert systems.

F. Uses of Expert Systems

The uses of expert systems are virtually limitless. They can be used to: diagnose, repair, monitor, analyse, interpret, consult, plan, design, instruct, explain, learn, and conceptualize.

G. Architecture of Expert Systems

One way to classify expert systems is by function (e.g. diagnosis, planning, etc). However, examination of existing expert systems indicates that there is little commonality in detailed system architecture that can be detected from this classification. A more fruitful approach appears to be to look at problem complexity and problem structure and deduce what data and control structures might be appropriate to handle these factors.

The Knowledge Engineering community has evolved a number of techniques (presented in the excellent tutorial by Stefik et al. (1982) and summarized in Gevarter (1982)) which can be utilized in devising suitable expert system architectures.

The use of these techniques in four existing expert systems is illustrated in Table I-1-1 thru I-1-4. Table I-1-1 thru I-1-4 outlines the basic approach taken by each of these expert systems and

*Not all expert systems are rule-based. The network-based expert systems MACSYMA, INTERNIST/CADUCEUS, Digitalis Therapy Advisor, HARPY and PROSPECTOR are examples which are not. Buchanan and Duda (1982) state that the basic requirements in the choice of an expert system knowledge representation scheme are extendibility, simplicity and explicitness. Thus, rule-based systems are particularly attractive.

TABLE I-1-1. Characteristics of Example Expert Systems.

SYSTEM: DENDRAL
INSTITUTION: Stanford University
AUTHORS: Feigenbaum & Lederberg
FUNCTION: Data Interpretation

		Key Elements of		
Purpose	**Approach**	**Knowledge Base**	**Global Data Base**	**Control Structure**
Generate plausible structural representations of organic molecules from mass spectrogram data	1. Derive constraints from the data. 2. Generate candidate structures. 3. Predict mass spectrographs for candidates. 4. Compare with data.	Rules for deriving constraints on molecular structure from experimental data Procedure for generating candidate structures to satisfy constraints Rules for predicting spectrographs from structures	Mass spectrogram data Constraints Candidate structures	Forward chaining Plan, generate and test.

TABLE I-1-2. Characteristics of Example Expert Systems.

SYSTEM: AM
INSTITUTION: Stanford University
AUTHORS: Lenat
FUNCTION: Concept Formation

		Key Elements of		
Purpose	**Approach**	**Knowledge Base**	**Global Data Base**	**Control Structure**
Discovery of mathematical concepts	Start with elementary ideas in set theory.	Elementary ideas in finite set theory.	Plausible candidate concepts.	Plan, generate, and test.
	Search a space of possible conjectures that can be generated from these elementary ideas.	Heuristics for generating new mathematical concepts by modifying and combining elementary ideas.		
	Choose the most interesting conjectures and pursue that line of reasoning.	Heuristics of "interestingness" for discarding bad ideas.		

TABLE I-1-3. Characteristics of Example Expert Systems.

SYSTEM: RI
INSTITUTION: CMU
AUTHORS: McDermott
FUNCTION: Design

		Key Elements of		
Purpose	**Approach**	**Knowledge Base**	**Global Data Base**	**Control Structure**
Configure VAX computer systems (from a customer's order of components).	Break problem up into the following ordered subtasks: 1. Correct mistakes in order. 2. Put components into CPU cabinets. 3. Put boxes into unibus cabinets and put components in boxes. 4. Put panels in unibus cabinets. 5. Lay out system on floor. 6. Do the cabling. Solve each subtask and move on to the next one in the fixed order.	Properties of (roughly 500) VAX components. Rules for determining when to move to next subtask based on system state. Rules for carrying out subtasks (to extend partial configuration). (Approximately 1200 rules total)	Customer order. Current task. Partial configuration (System state).	"MATCH" (data driven) (no backtracking)

TABLE I-1-4. Characteristics of Example Expert Systems.

SYSTEM: MYCIN
INSTITUTION: Stanford University
AUTHORS: Shortliffe
FUNCTION: Diagnosis

Purpose	Approach	Key Elements of: Knowledge Base	Key Elements of: Global Data Base	Key Elements of: Control Structure
Diagnosis of bacterial infections and recommendations for antibiotic therapy.	Represent expert judgmental reasoning as condition-conclusion rules together with the expert's "certainty" estimate for each rule.	Rules linking patient data to infection hypotheses.	Patient history and diagnostic tests.	Backward chaining thru the rules.
			Current hypothesis.	Exhaustive search.
	Chain backwards from hypothesized diagnoses to see if the evidence supports it.	Rules for combining certainty factors.	Status.	
	Exhaustively evaluate all hypotheses.	Rules for treatment.	Conclusions reached thus, far, and rule numbers justifying them.	
	Match treatments to all diagnoses which have high certainty values.			

shows how the approach translates into key elements of the Knowledge Base, Global Data Base and Control Structure. An indication of the basic control structures of the systems in Table I-1-1 thru I-1-4, and some of the other well known expert systems, is given in Table I-2.

Table I-2 represents expert system control structures in terms of the search direction, the control techniques utilized, and the search space transformations employed. The approaches used in the various expert systems are different implementations of two basic ideas for overcoming the combinatorial explosion associated with search in real complex problems. These two ideas are:

(1) Find ways to efficiently search a space,
(2) Find ways to transform a large search space into smaller manageable chunks that can be searched efficiently.

It will be observed from Table I-2 that there is little architectural commonality based either on function or domain of expertise. Instead, expert system design may best be considered as an art form, like custom home architecture, in which the chosen design can be implemented from the collection of available AI techniques in heuristic search and problem solving.

In addition to the techniques indicated in Table I-2, also emerging are distributed knowledge and problem solving approaches exemplified by the MDX expert system (Chandrasekaran, 1983) and the object-oriented programming language, LOOPS (Stefik et al., 1983).

H. Existing Expert Systems

Table I-3 is a list, classified by function and domain of use, of most of the existing major expert systems. It will be observed that there is a predominance of systems in the Medical and Chemistry domains following from the pioneering efforts at Stanford University. From the list, it is also apparent that Stanford University dominates in number of systems, followed by M.I.T., CMU, BBN and SRI, with several dozen scattered efforts elsewhere.

The list indicates that thus far the major areas of expert systems development have been in diagnosis, data analysis and interpretation, planning, computer-aided instruction, analysis, and automatic programming. However, the list also indicates that a number of pioneering expert systems already exist in quite a number of other functional areas. In addition, a substantial effort is under way to build expert systems as tools for constructing expert systems.

I. Constructing an Expert System

Duda (1981, p. 262) states that to construct a successful expert system, the following prerequisites must be met:

- there must be at least one human expert acknowledged to perform the task well.
- the primary source of the expert's exceptional performance must be special knowledge, judgment, and experience.
- the expert must be able to explain the special knowledge and experience and the methods used to apply them to particular problems.
- the task must have a well-bounded domain of application.

Using present techniques and programming tools, the effort required to develop an expert system appears to be converging towards five man-years, with most endeavors employing two to five people in the construction.

TABLE I-2. Control Structures of Some Well Known Expert Systems.

			Control Structure																
			Search Direction				Control								Search Space Transformations				
System	Function	Domain	Forward	Backward	Forward and Backward	Event Driven	Exhaustive Search	Generate and Test	Guessing	Relevant Backtracking	Least Commitment	Multilines of Reasoning	Network Editor	Beam Search	Multiple Models	Break into Sub-Problems	Hierarchical Refinement	Hierarchical Resolution	Meta Rules
MYCIN	Diagnosis	Medicine		x			x												
DENDRAL	Data Interpr.	Chemistry	x					x											
EL	Analysis	Elec. Circuits	x						x	x									
GUIDON	C.A.I.	Medicine				x													
KAS	Knowl. Acquis.	Geology	x										x						
META-DENDRAL	Learning	Chemistry	x					x											
AM	Concept Formation	Math	x					x											
VM	Monitoring	Medicine				x	x												
GA1	Data Interpr.	Chemistry	x					x											
R1	Design	Computers	x				x									x			
ABSTRIPS	Planning	Robots		x													x		
NOAH	Planning	Robots		x							x					x			
MOLGEN	Design	Genetics			x				x	x	x					x	x		x
SYN	Design	Elec. Circuits	x												x				
HEARSAY II	Signal Interpr.	Speech Unders.			x						x	x						x	
HARPY	Signal Interpr.	Speech Unders.	x											x					
CRYSALIS	Data Interpr.	Crystallography				x		x											x

TABLE I-3. Existing Expert Systems by Function.

Function	Domain	System*	Institution
Diagnosis	Medicine	PIP	M.I.T.
	Medicine	CASNET	Rutgers U.
	Medicine	INTERNIST/CADUCEUS	U. of Pittsburgh
	Medicine	MYCIN	Stanford U.
	Medicine	PUFF	Stanford U.
	Medicine	MDX	Ohio State U.
	Computer Faults	DART	Stanford U./IBM
	Computer Faults	IDT	DEC
	Nuclear Reactor Accidents	REACTOR	E G & G Idaho Inc.
Data Analysis and Interpretation	Geology	DIPMETER ADVISOR	M.I.T./Schlumberger
	Chemistry	DENDRAL	Stanford U.
	Chemistry	GA1	Stanford U.
	Geology	PROSPECTOR	SRI
	Protein Crystallography	CRYSALIS	Stanford U.
	Determination of Causal Relationships in Medicine	RX	Stanford U.
	Determination of Causal Relationships in Medicine	ABEL	M.I.T.
	Oil Well Logs	ELAS	AMOCO
Analysis	Electrical Circuits	EL	M.I.T.
	Symbolic Mathematics	MACSYMA	M.I.T.
	Mechanics Problems	MECHO	Edinburgh
	Naval Task Force Threat Analysis	TECH	Rand/NOSC
	Earthquake Damage Assessment for Structures	SPERIL	Purdue U.
	Digital Circuits	CRITTER	Rutgers U.
Design	Computer System Configurations	R1/XCON	C.M.U./DEC
	Circuit Synthesis	SYN	M.I.T.
	Chemical Synthesis	SYNCHEM	SUNY Stonybrook

*References to these systems can be found in Duda (1981), Stefik, et al. (1982), Buchanan (1981), Buchanan and Duda (1982), Barr and Feigenbaum (1982), IJCAI-81, and AAAI-82.

TABLE I-3. Existing Expert Systems by Function. (cont.)

Function	Domain	System*	Institution
Planning	Chemical Synthesis	SECHS	U. of Cal. Santa Cruz
	Robotics	NOAH	SRI
	Robotics	ABSTRIPS	SRI
	Planetary Flybys	DEVISER	JPL
	Errand Planning	OP-PLANNER	Rand
	Molecular Genetics	MOLGEN	Stanford U.
	Mission Planning	KNOBS	MITRE
	Job Shop Scheduling	ISIS-II	CMU
	Design of Molecular Genetics Experiments	SPEX	Stanford U.
	Medical Diagnosis	HODGKINS	M.I.T.
	Naval Aircraft Ops	AIRPLAN	CMU
	Tactical Targeting	TATR	RAND
Learning from Experience	Chemistry	METADENDRAL	Stanford U.
	Heuristics	EURISKO	Stanford U.
Concept Formation	Mathematics	AM	CMU
Signal Interpretation	Speech Understanding	HEARSAY II	CMU
	Speech Understanding	HARPY	CMU
	Machine Acoustics	SU/X	Stanford U.
	Ocean Surveillance	HASP	System Controls Inc.
	Sensors On Board Naval Vessels	STAMMER-2	NOSC, San Diego/SDC
	Medicine—Left Ventrical Performance	ALVEN	U. of Toronto
	Military Situation Determination	ANALYST	MITRE
Monitoring	Patient Respiration	VM	Stanford U.
Use Advisor	Structural Analysis Computer Program	SACON	Stanford U.
Computer Aided Instruction	Electronic Troubleshooting	SOPHIE	B.B.N.
	Medical Diagnosis	GUIDON	Stanford U.
	Mathematics	EXCHECK	Stanford U.
	Steam Propulsion Plant Operation	STEAMER	BBN
	Diagnostic Skills	BUGGY	BBN
	Causes of Rainfall	WHY	BBN
	Coaching of a Game	WEST	BBN
	Coaching of a Game	WUMPUS	M.I.T.
		SCHOLAR	BBN

TABLE I-3. Existing Expert Systems by Function. (cont.)

Function	Domain	System*	Institution
Knowledge Acquisition	Medical Diagnosis	TEIRESIAS	Stanford U.
	Medical Consultation	EXPERT	Rutgers
	Geology	KAS	SRI
Expert System Construction		ROSIE	Rand
		AGE	Stanford U.
		HEARSAY III	USC/ISI
		EMYCIN	Stanford U.
		OPS 5	CMU
		RAINBOW	IBM
	Medical Diagnosis	KMS	U. of MD
	Medical Consultation	EXPERT	Rutgers
	Electronic Systems Diagnosis	ARBY	Smart Sys. Tech.
	Medical Consultation Using Time-Oriented Data	MECS-AI	Tokyo U.
Consultation/Intelligent Assistant	Battlefield Weapons Assignments	BATTLE	NRL AI Lab
	Medicine	Digitalis Therapy Advisor	M.I.T.
	Radiology	RAYDEX	Rutgers U.
	Computer Sales	XCEL	CMU/DEC
	Medical Treatment	ONCOCIN	Stanford U.
	Nuclear Power Plants	CSA Model-Based Nuclear Power Plant Consultant	GA Tech
	Diagnostic Prompting in Medicine	RECONSIDER	U. of CA, S.F.
Management	Automated Factory	IMS	CMU
	Project Management	CALLISTO	DEC
Automatic Programming	Modelling of Oil Well Logs	ΦNIX	Schlumberger-Doll Res.
		CHI	Kestrel Inst.
		PECOS	Stanford U.
		LIBRA	Stanford U.
		SAFE	USC/ISI
		DEDALUS	SRI
		Programmer's Apprentice	M.I.T.
Image Understanding		VISIONS	U. of Mass.
		ACRONYM	Stanford U.

J. Summary of the State-of-the-Art

Buchanan (1981, pp. 6-7) indicates that the current state of the art in expert systems is characterized by:

- *Narrow domain of expertise*

Because of the difficulty in building and maintaining a large knowledge base, the typical domain of expertise is narrow. The principal exception is INTERNIST, for which the knowledge base covers 500 disease diagnoses. However, this broad coverage is achieved by using a relatively shallow set of relationships between diseases and associated symptoms. (INTERNIST is now being replaced by CADUCEUS, which uses causal relationships to help diagnose simultaneous unrelated diseases.)

- *Limited knowledge representation languages for facts and relations*

- *Relatively inflexible and stylized input-output languages*

- *Stylized and limited explanations by the systems*

- *Laborious construction*

At present, it requires a knowledge engineer to work with a human expert to laboriously extract and structure the information to build the knowledge base. However, once the basic system has been built, in a few cases it has been possible to write knowledge acquisition systems to help extend the knowledge base by direct interaction with a human expert, without the aid of a knowledge engineer.

- *Single expert as a "knowledge czar."*

We are currently limited in our ability to maintain consistency among overlapping items in the knowledge base. Therefore, though it is desirable for several experts to contribute, one expert must maintain control to insure the quality of the data base.

- *Fragile behavior*

In addition, most systems exhibit fragile behavior at the boundaries of their capabilities. Thus, even some of the best systems come up with wrong answers for problems just outside their domain of coverage. Even within their domain, systems can be misled by complex or unusual cases, or for cases for which they do not yet have the needed knowledge or for which even the human experts have difficulty.

- *Requires Knowledge Engineer to Operate*

Another limitation is that for most current systems only their builders or other knowledge engineers can successfully operate them - a friendly interface not having yet been constructed.

Nevertheless, Randy Davis (1982) observes that there have been notable successes. A methodology has been developed for explicating informal knowledge. Representing and using empirical associations, five systems have been routinely solving difficult problems — DENDRAL, MACSYMA, MOLGEN, R1 and PUFF — and are in regular use. The first three all have serious users who are only loosely coupled to the system designers. DENDRAL, which analyzes chemical instrument data to determine the underlying molecular structure, has been the most widely used program (see Lindsay et al., 1980). R1, which is used to configure VAX computer systems, has been reported to be saving DEC twenty million dollars per year, and is now being followed up with XCON. In addition, as indicated in Table I-3, dozens of systems have been constructed and are being experimented with.

K. Future Trends

Figure I-2 lists some of the expert systems applications currently under development.

It will be observed that there appear to be few domain or functional limitations in the ultimate use of expert systems. However, the nature of expert systems is changing. The limitations of rule-based systems are becoming apparent. Not all knowledge can be readily structured in the form of empirical associations. Empirical associations tend to hide causal relations (present only implicitly in such associations). Empirical associations are also inappropriate for highlighting structure and function.

Thus, the newer expert systems are adding deep knowledge having to do with causality and structure. These systems will be less fragile, thereby holding the promise of yielding correct answers often enough to be considered for use in autonomous systems, not just as intelligent assistants.

The other change is a trend towards an increasing number of non-rule based systems. These systems, utilizing semantic networks, frames and other knowledge representations, are often better suited for causal modeling and representing structure. They also tend to simplify the reasoning required by providing knowledge representations more appropriate for the specific problem domain.

- Medical diagnosis and prescription
- Medical knowledge automation
- Chemical data interpretation
- Chemical and biological synthesis
- Mineral and oil exploration
- Planning/scheduling
- Signal interpretation
- Signal fusion—situation interpretation from multiple sensors
- Military threat assessment
- Tactical targeting
- Space defense
- Air traffic control
- Circuit diagnosis
- VLSI design
- Equipment fault diagnosis
- Computer configuration selection
- Speech understanding
- Intelligent Computer-Aided Instruction
- Automatic Programming
- Intelligent knowledge base access and management
- Tools for building expert systems

Figure I-2. Expert System Applications Now Under Development.

Figure I-3 (based largely on Hayes-Roth IJCAI-81 Expert system tutorial and on Feigenbaum, 1982) indicates some of the future opportunities for expert systems. Again no limitation is apparent.

It thus appears that expert systems will eventually find use in most endeavors which require symbolic reasoning with detailed professional knowledge — which includes much of the world's work. In the process, there will be exposure and refinement of the previously private knowledge in the various fields of applications.

On a more near-term scale, in the next few years we can expect to see expert systems with thousands of rules. In addition to the increasing number of rule-based systems we can also expect to see an increasing number of non-rule based systems. Also anticipated are much improved ex-

- *Building and Construction*
  Design, planning, scheduling, control
- *Equipment*
  Design, monitoring, control, diagnosis, maintenance, repair, instruction.
- *Command and Control*
  Intelligence analysis, planning, targeting, communication
- *Weapon Systems*
  Target identification, adaptive control, electronic warfare
- *Professions*
  (Medicine, law, accounting, management, real estate, financial, engineering)
  Consulting, instruction, analysis
- *Education*
  Instruction, testing, diagnosis, concept formation and new knowledge development from experience.
- *Imagery*
  Photo interpretation, mapping, geographic problem-solving.
- *Software*
  Instruction, specification, design, production, verification, maintenance
- *Home Entertainment and Advice-giving*
  Intelligent games, investment and finances, purchasing, shopping, intelligent information retrieval
- *Intelligent Agents*
  To assist in the use of computer-based systems
- *Office Automation*
  Intelligent systems
- *Process Control*
  Factory and plant automation
- *Exploration*
  Space, prospecting, etc.

Figure I-3. Future Opportunities for Expert Systems.

planation systems that can explain (make "transparent") why an expert system did what it did and what things are of importance.

By the late 80's, we can expect to see intelligent, friendly and robust human interfaces and much better system building tools.

Somewhere around the year 2000, we can expect to see the beginnings of systems which semi-autonomously develop knowledge bases from text. The result of these developments may very well herald a maturing information society where expert systems put experts at everyone's disposal. In the process, production and information costs should greatly diminish, opening up major new opportunities for societal betterment.

REFERENCES

• Barr, A., and Feigenbaum, E. A., *The Handbook of Artificial Intelligence,* Vol. 2, Los Altos, CA: W. Kaufman, 1982.

• Buchanan, B. G., "Research on Expert Systems," Stanford University Computer Science Department, Report No. STAN-CS-81-837, 1981.

• Buchanan, B. G. and Duda, R. O., "Principles of Rule-Based Expert Systems," Heuristic Programming Project Report No. HPP 82-14, Dept. of Computer Science, Stanford, CA, Aug. 1982. (To appear in *Advances in Computers, Vol. 22,* M. Yorit. (ed.), New York: Academic Press)

• Chandrasekaran, B., "Towards a Taxonomy of Problem Solving Types," *The AI Magazine,* Vol 4, No. 1, Winter-Spring 1983, pp 9–17.

• Davis, R., "Expert Systems: Where Are We: and Where Do We Go From Here?" *AI Magazine,* Vol. 3. No. 2, Spring 82, pp. 3-25.

• Duda, R. O. "Knowledge-Based Expert Systems Come of Age," *Byte,* Vol. 6, No. 9, Sept. '81 pp. 238-281.

• Feigenbaum, E. A., "Knowledge Engineering for the 1980's," Computer Science Dept., Stanford University, 1982.

• Gevarter, W. B., *An Overview of Expert Systems,* NBSIR 82-2505, National Bureau of Standards, Washington, D.C., May 1982 (Revised October 1982).

• Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J., *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project,* New York: McGraw-Hill, 1980.

• Stefik, M., Alkins, J., Balzer, R., Benoit, J., Birnbaum, L. Hayes-Roth, R., Sacerdoti, E., "The Organization of Expert Systems, A Tutorial, "*Artificial Intelligence,* Vol. 18, 1982, pp. 135-173.

• Stefik, M., Bobrow, D. G., Mittal, S., and Conway, L., "Knowledge Programming in LOOPS," *The AI Magazine,* Vol 4, No. 3, Fall 1983, pp 3–13.

• *IJCAI-81 — Proc. of The International Joint Conference on AI,* Vancouver, Aug. 1981.

• *AAAI-82 — Proc. of the National Conference on AI,* CMU & U. of Pittsburgh, Pittsburgh, PA, Aug. 18-22, 1982.

• *IJCAI-83 — Proc. of The Eighth International Joint Conference on AI,* Karlsruhe, W. Germany, Aug. 1983.

• *AAAJ-83 — Proc. of The National Conference on AAAI,* Wash, D.C., Aug. 1983.

II. COMPUTER VISION

A. Introduction

Computer Vision — visual perception employing computers — shares with "Expert Systems" the role of being one of the most popular topics in Artificial Intelligence today. The computer vision field is multifaceted, having many participants with diverse viewpoints, with many papers having been written. However, the field is still in the early stages of development — organizing principles have not yet fully crystalized, and the associated technology has not yet been completely rationalized. However, commercial vision systems have already begun to be used in manufacturing and robotic systems for inspection and guidance tasks, and other systems (at various stages of development) are beginning to be employed in military, cartographic and image interpretation applications.

B. Definition

Computer (computational or machine) vision can be defined as perception by a computer based on visual sensory input. Barrow and Tenenbaum (1981, p. 573) state:

> Vision is an information-processing task with well-defined input and output. The input consists of arrays of brightness values, representing projections of a three-dimensional scene recorded by a camera or comparable imaging device. Several input arrays may provide information in several spectral bands (color) or from multiple viewpoints (stereo or time sequence). The desired output is a concise description of the three-dimensional scene depicted in the image, the exact nature of which depends upon the goals and expectations of the observer. It generally involves a description of objects and their interrelationships, but may also include such information as the three-dimensional structures of surfaces, their physical characteristics (shape, texture, color, material), and the locations of shadows and light sources . . .

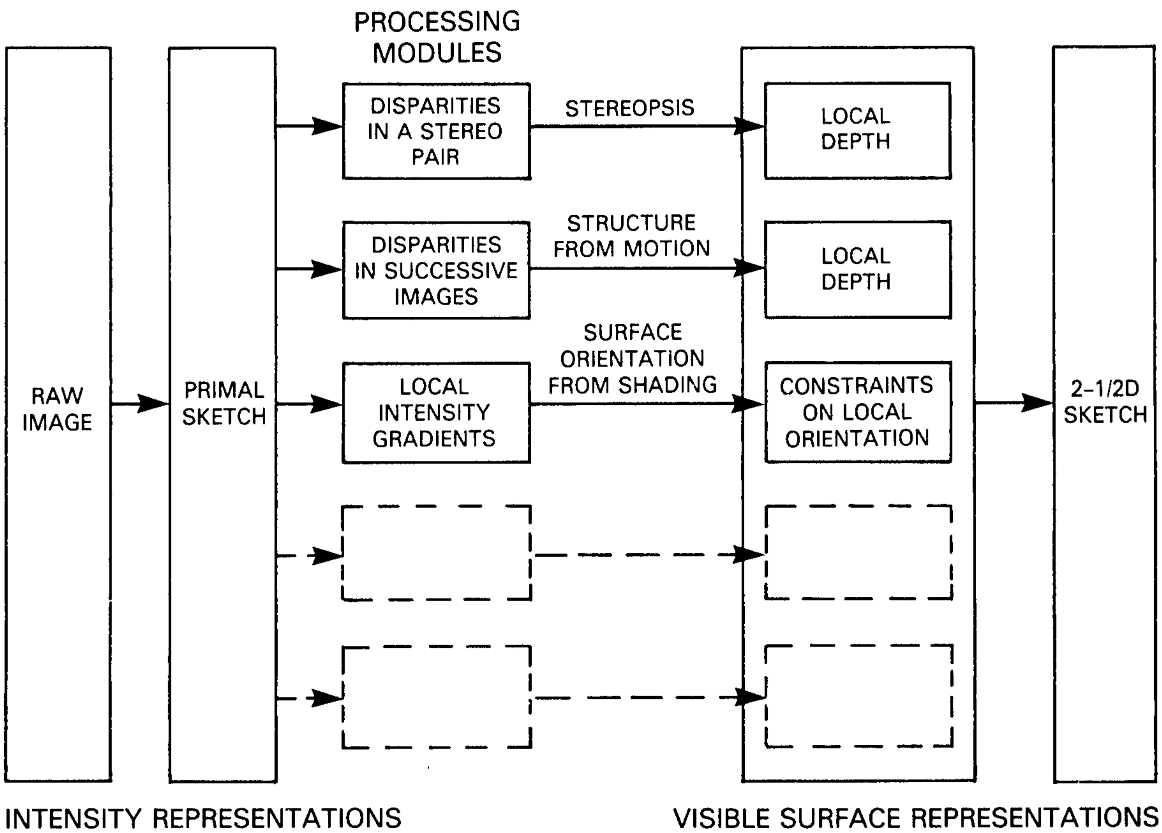
C. Relation to Human Vision

MIT's Marr and Nishihara (1978, p. 42) take the view that "Artificial Intelligence is (or ought to be) the study of information processing problems that characteristically have their roots in some aspects of biological information processing." They developed a computational theory of vision based on their study of human vision. Figure II-1 represents the transition from the raw image through the primal sketch to the 2-1/2D sketch (exemplified by Figure II-2), which contains information on local surface orientations, boundaries, and depths.

The primal sketch, reminiscent of an artist's hurried drawing, is a primitive but rich description of the way the intensities change over the visual field. It can be represented by a set of short line segments separating regions of different brightnesses. A list of the properties of the lines segments, such as location, length, and orientation for each segment can be used to represent the primal sketch.

The late Dr. Marr and his associates' development of a human visual information processing theory (Marr, 1982) has had a substantial impact on computational vision.

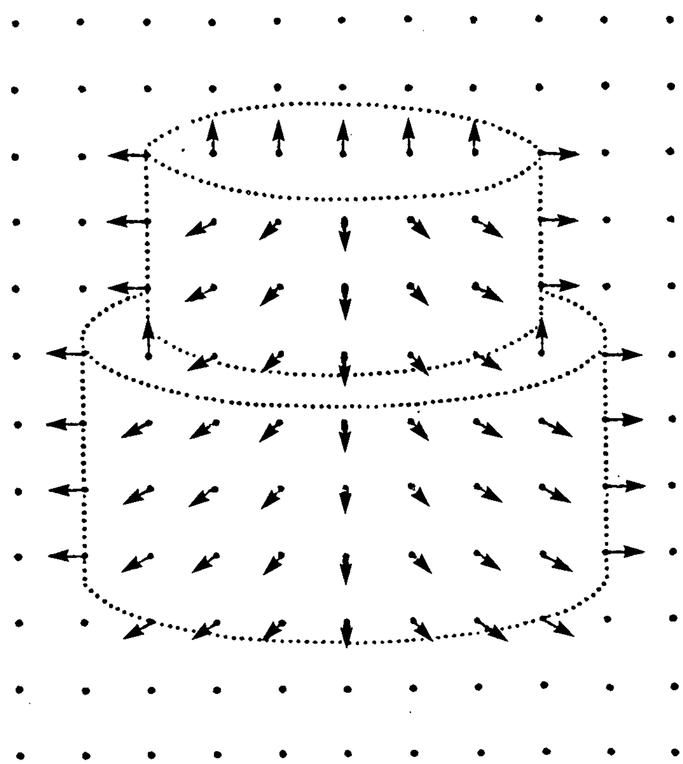
There are strong indications (see, e.g., Gevarter, 1977) that the interpretative planning areas of the human brain set up a context for processing the input data. (This viewpoint is captured by



The computations begin with representations of the intensities in an image—first the image itself, (e.g., the gray-level intensity array) and then the primal sketch, a representation of spatial variations in intensity. Next comes the operation of a set of modules, each employing certain aspects of the information contained in the image to derive information about local orientation, local depth, and the boundaries of surfaces. From this is constructed the so-called 2-1/2 dimensional sketch. Note that no "high-level" information is yet brought to bear: the computations proceed by utilizing only what is available in the image itself.

After: Marr and Nishihara, 1978, p. 42.

Figure II-1. A Framework for Early and Intermediate States in A Theory of Visual Information Processing.



A candidate for the so-called 2-½-dimensional sketch, which encompasses local determinations of the depth and orientation of surfaces in an image, as derived from processes that operate upon the primal sketch or some other representation of changes in gray-level intensity. The lengths of the needles represent the degree of tilt at various points in the surface; the orientations of the needles represent the directions of tilt. . . Dotted lines show contours of surface discontinuity. No explicit representation of depth appears in this figure.

Source: Marr and Nishihara, 1978, p. 41.

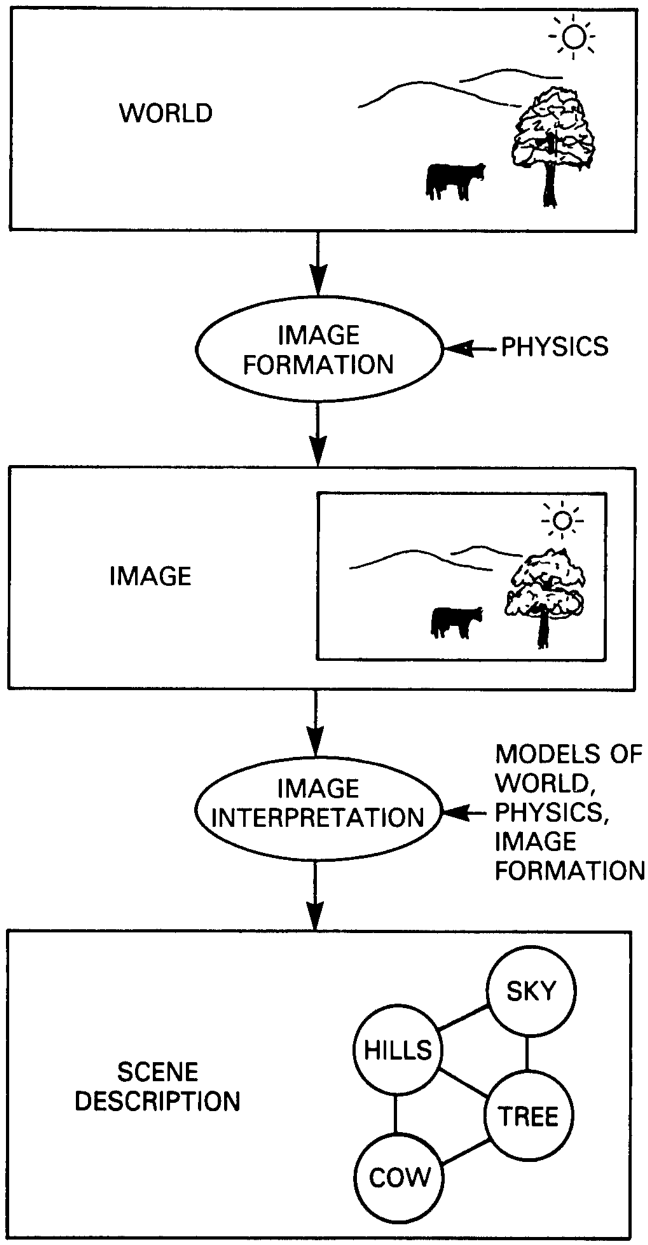
Figure II-2. An Example of a 2-1/2D Sketch.

Minsky's (1975) AI "frame" concept for knowledge representation.) The brain then uses visual and other cues from the environment to draw in past knowledge to generate an internal representation and interpretation of the scene. This knowledge-based expectation-guided approach to vision is now appearing in advanced AI computer vision systems.

D. Basis for a General Purpose Image Understanding System

Barrow and Tenenbaum (1981, p. 573) observe that in going from a scene to an image (an array of brightness values) that the image encodes much information about the scene, but the information is confounded in the single brightness value at each point. In projecting onto the two-dimensional image, information about the three-dimensional structure of the scene is lost. In order to decode brightness values and recover a scene description, it is necessary to employ a priori knowledge embodied in models of the scene domain, the illumination, and the imaging process.

As indicated by Figure II-3, computer vision is an active process that uses these models to interpret the sensory data. To accommodate the diversity of appearance found in real imagery, a high-performance, general-purpose system must embody a great deal of knowledge in its models.



Source: Barrow and Tenenbaum, 1981, p. 573.

Figure II-3. Model-based Interpretation of Images.

E. Basic Paradigms for Computer Vision*

In broad terms, an image understanding system starts with the array of pixel amplitudes that define the computer image, and using stored models (either specific or generic) determines the content of a scene. Typically, various symbolic features such as lines and areas are first determined from the image. These are then compared with similar features associated with stored models to find a match, when specific objects are being sought. In more generic cases, it is necessary to determine various characteristics of the scene, and using generic models determine from geometric shapes and other factors (such as allowable relationships between objects) the nature of the scene content.

A variety of paradigms have been proposed to accomplish these tasks in image understanding systems. These paradigms are based on a common set of broadly defined processing and manipulating elements: feature extraction, symbolic representation, and semantic interpretation. The paradigms differ primarily in how these elements (defined below) are organized and controlled, and the degree of artificial intelligence and knowledge employed.

1. Hierarchical Bottom-up Approach

Figure II-4A is a block diagram of a hierarchical paradigm of an image understanding system that employs a bottom-up processing approach. The hierarchical bottom-up approach can be developed successfully for domains with simple scenes made up of only a limited number of previously known objects.

2. Hierarchical Top-down Approach

This approach (usually called hypothesize and test), shown in Figure II-4B, is goal directed, the interpretation stage being guided in its analysis by trial or test descriptions of a scene. An example would be using template matching — matched filtering — to search for a specific object or structure within the scene. Matched filtering is normally performed at the pixel level by cross correlation of an object template with an observed image field. It is often computationally advantageous, because of the reduced dimensionality, to perform the interpretation at a higher level in the chain by correlating image features or symbols rather than pixels.

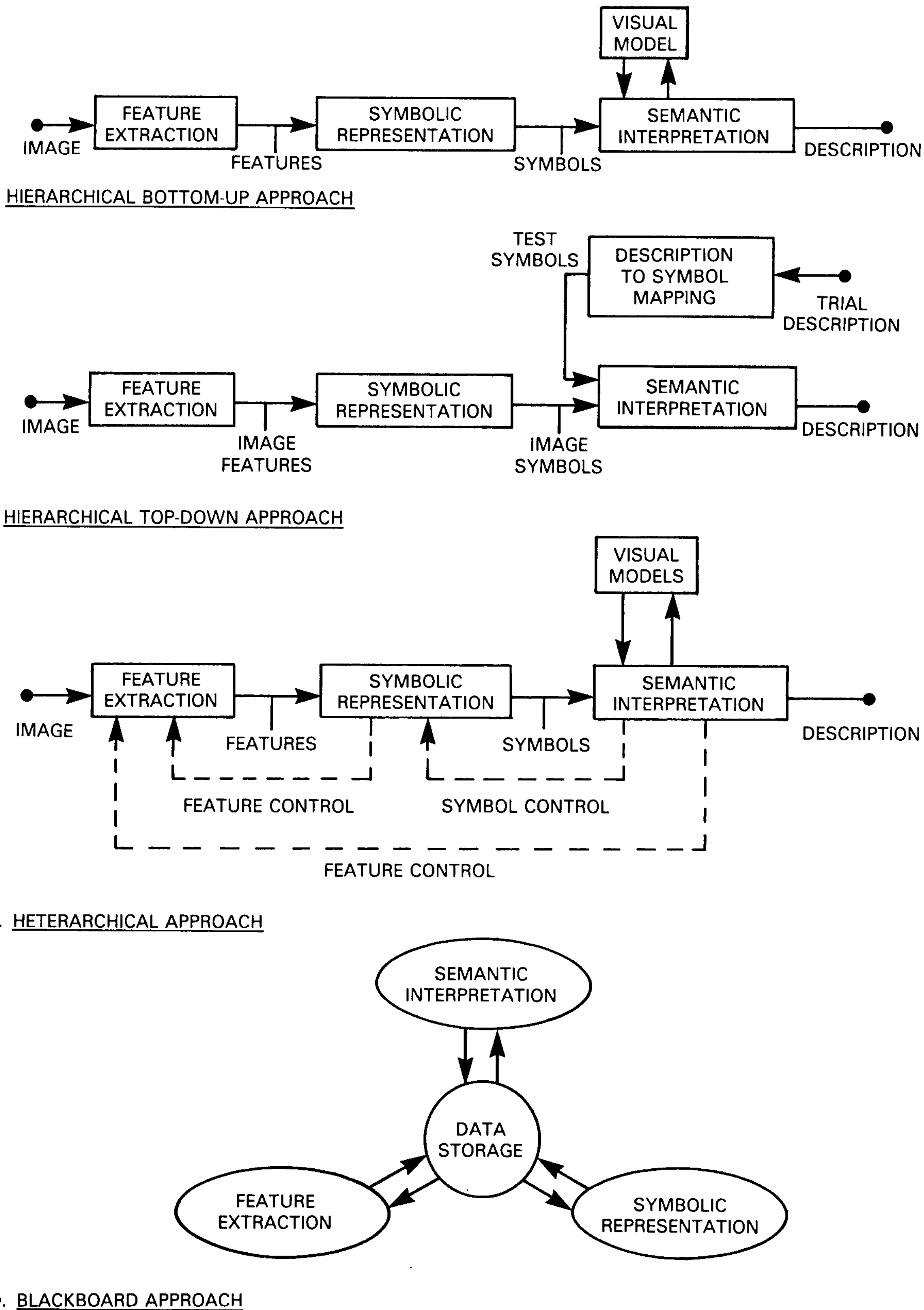
3. Heterarchical Approach

Hierarchical image understanding systems are normally designed for specific applications. They thus tend to lack adaptability. A large amount of processing is also usually required. Pratt (1978) (pp. 572-573) observes that often much of this processing is wasted in the generation of features and symbols not required for the analysis of a particular scene. A technique to avoid this problem is to establish a central monitor to observe the overall performance of the image understanding system and then issue commands to the various system elements to modify their operation to maximize system performance and efficiency.

Figure II-4C is a block diagram of an image understanding system that achieves heterarchical operation by distributed feedback control.

---

*This section is primarily based on Pratt, 1978, pp. 570-574.



Source: Pratt, 1978, pp. 570-574.

Figure II-4. Basic Image Understanding Paradigms.

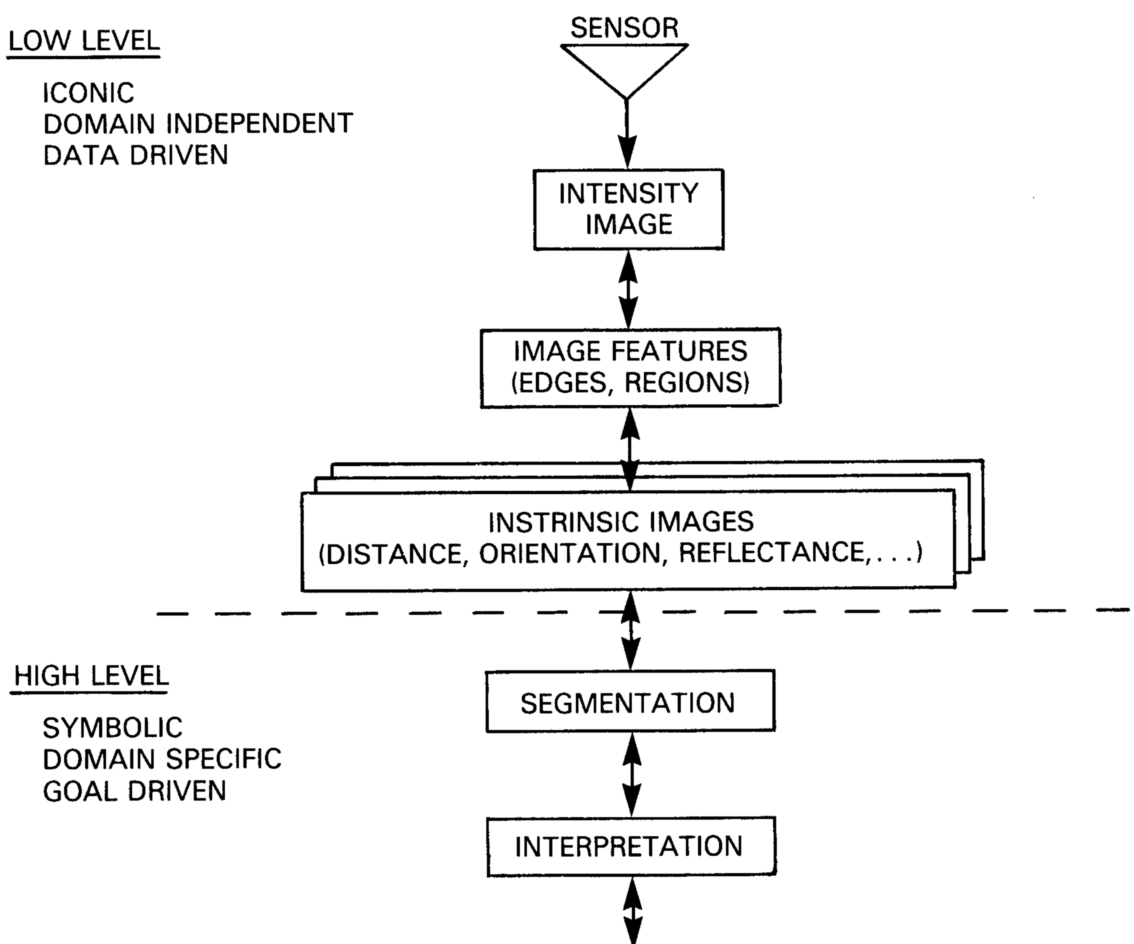
4. *Blackboard Approach*

Another image understanding system configuration called the blackboard model has been proposed by Reddy and Newell (1975). Figure II-4D is a simplified representation of this approach in which the various system elements communicate with each other via a common working data storage called the blackboard. Whenever any element performs a task, its output is put into the common data storage, which is independently accessible by all other elements. The individual elements can be designed to act autonomously to further the common system goal as required. The blackboard system is particularly attractive in cases where several hypotheses must be considered simultaneously and their components need to be kept track of at various levels of representation.

F. Levels of Representation

A computer vision system, like human vision is, commonly considered to be naturally structured as a succession of levels of representation.

Tenenbaum, et al. (1979, pp. 254-255), sketch in Figure II-5, a way in which to view an organization of a general-purpose vision system. They divide the figure into two parts. The first is



Source: Tenenbaum et al., 1979, p. 255.

Figure II-5. Organization of a Visual System.

image oriented (iconic), domain independent, and based on the image data (data driven). The second part of the figure is symbolic, dependent on the domain and the particular goal of the vision process.

The first portion takes the image, which consists of an intensity array of picture elements (''pixels,'' e.g., 1000 $\times$ 1000), and converts it into image features such as edges and regions. These are then converted into a set of parallel ''intrinsic images,'' one each for distance (range), surface orientation, reflectance,* etc.

The second part of the system segments these into volumes and surfaces dependent on our knowledge of the domain and the goal of the computation. Using domain knowledge and the constraints associated with the relations among objects in this domain, objects are identified and the scene analyzed consistent with the system goal.

G. Research in Model-Based Vision Systems

Most research efforts in vision have been directed at exploring various aspects of vision, or toward generating particular processing modules for a step in the vision process rather than in devising general purpose vision systems. However, there are currently two major U.S. efforts in general purpose vision systems. The ACRONYM system at Stanford University under the leadership of T. Binford, and the VISIONS system at the University of Massachusetts at Amherst under A. Hanson and E. Riseman.

The ACRONYM system, outlined in Table II-1-1, is designed to be a general purpose, model-based system that does its major reasoning at the level of volumes rather than images. The system basically takes a hierarchical top-down approach as in Figure II-4B. ACRONYM has four essential parts: modeling, prediction, description and interpretation. The user provides ACRONYM with models of objects (modeled in terms of volume primitives called generalized cones) and their spatial relationships; as well as generic models and their subclass relationships. These are both stored in graph form. The program automatically predicts which image features to expect. Description is a bottom-up process that generates a model-independent description of the image. Interpretation relates this description to the prediction to produce a three-dimensional understanding of the scene.

The VISIONS system outlined in Table II-1-2, can be considered to be a working tool to test various image understanding modules and approaches. Rather than using specific models, its high level knowledge is in the form of framelike ''schemas'' which represent expectations and expected relationships in particular scene situations. VISIONS is based on monocular images and does its reasoning at the level of images rather than volumes.

Other research efforts in model-based vision systems are summarized in TABLES III in Appendix I of Gevarter (1982A). All the research computer vision systems are individually crafted by the developers — reflecting the developers' backgrounds, interests and domain requirements. All, except ACRONYM (and to an extent, 3-D Mosaic, Kanade, 1981), use image (2-D) models and are viewpoint dependent. Models are mostly described by semantic networks though feature vectors are also utilized. The systems, capitalizing on their choice to limit their observations to only a few

*Fraction of normal incident illumination reflected.

TABLE II-1-1. Model-Based Vision Systems.

Developer: Brooks et al. (1979), Brooks (1981)
System: ACRONYM
Purpose: General Purpose Vision System
Example Domains: Identifying Airplanes on a Runway in Aerial Images
Simulation for Robot Systems and for Automated Grasping of Objects

Approach	Modeling	Image Feature Extraction & Representation	Search & Matching	Remarks
Hierarchical top down appoach.	Represents object classes from which subclasses and specific objects are represented by numeric constraints.	Ribbons and curves obtained from an edge mapper.	Matcher does an interpretation matching by mapping the Observability Graph into the Picture Graph.	Aims to be a general vision system.
Reasons between different levels of representation based on a hierarchy of representations.		Surfaces obtained from a stereo mapper.		Insensitive to viewpoint.
High level modeler provides a high level language to manipulate models using symbolic names.	Models 3-D objects using volume primitives: generalized cones and ribbons.	Nodes of the Picture Graph (symbolic version of image) correspond to ribbons, surfaces and curves.	Matcher works in a coarse to fine order.	A goal is to make use of total information for interpretation.
Predictor and Planner Module is a rule-based system to generate an Observability Graph from the Object Graph (3-D object representation consisting of nodes and relational arcs).	Spatial relations of volume elements within an object defined hierarchically.	Arcs and relations indicate spatial relations between nodes.	Combines local matches of ribbons into clusters.	Feature extraction (e.g., finding lines and regions) still weak.
Makes predictions (which are viewpoint insensitive) in the form of symbolic constraint expressions with variables.	Can model both specific and generic volume elements and relations between them.		Searches for maximal subgraph matches in the Observability graph.	Interpretation is limited to scenes with few objects.
Makes a projective transformation from models.	Models are part/whole graphs.		Performs major interpretation at the level of volumes rather than at the level of images.	Substantial progress has been achieved in past few years.
Predicts appearances of models in images in terms of ribbons and ellipses.	Volume primitives have local rather than viewer-centered primitives.			

TABLE II-1-1. Model-Based Vision Systems (cont.)

Approach	Modeling	Image Feature Extraction & Representation	Search & Matching	Remarks
Incorporates translation and rotation into observable representations.				
Searches for instances of models in images. It employs geometric reasoning in the form of a rule based problem-solving system.				
It interprets (matches) in 3-D by enforcing constraints of the 3-D model.				

TABLE II-1-2. Model-Based Vision Systems.

Developer: Hanson & Riseman (1978a,b)
Systems: VISIONS
Purpose: Interpreting static monocular scenes
Can be considered to be a working tool to test various image understanding modules and approaches
Example Domains: House scenes from ground level
Road scenes from ground level

Approach	Modeling	Image Feature Extraction & Representation	Search & Matching	Remarks
Uses hierarchical modular approach to representation and control.	Hierarchical structure	Uses both edge finding and region growing to segment the image into a layered directed graph of regions, line segments and vertices	Generates and stores partial models in "contexts" (of the CONNIVER programming language) which provide a history of decisions to be used when backtracking is necessary	System (Parma, 1980) did reasonably well in making a crude segmentation of a house scene
Tries to be as general as possible to allow both bottom-up and top-down solution hypotheses as well as various intermediate combinations	Scene schemas (like frames) are the highest representation			
	Hierarchy is: —schemas —objects —volumes —surfaces	Uses a hierarchical processing cone (pyramid) to be able to handle image data at various levels of resolution		Viewpoint dependent
Incorporates the flexibility to utilize various feature extraction modules and multiple knowledge sources as required	Proposed representations of 3D surfaces and volumes include: —generalized cylinders —surface patches with cubic B-splines to represent boundary and blending functions	Uses a relaxation approach to organize edges into boundaries, and pixel clusters into regions, using high-level system guidance (interpretation guided segmentation)	Uses a multiple knowledge source heterarchical approach which generates partial models in the search space of models. Attempts, using top-down and bottom-up relaxation techniques, to converge on a most probable solution.	Schema used depends on specific scene.
Allows for the possibility of generating and verifying hypotheses along many paths	Employs semantic networks —nodes represent primitive entities (objects, concepts situations, etc.) —Labeled arcs represent relationships between them		Uses rules for *focusing* on an element of a task, *expanding* that element by generating new hypotheses and *verifying* new hypotheses.	

objects, use predominantly the top-down interpretation of images approach, relying heavily on prediction.

H. Industrial Vision Systems

1. General Characteristics

The prominent aspect of industrial vision systems, in distinction to more general vision systems, is that they operate in a relatively known and structured environment. In addition, the situation (such as placement of cameras and lighting) can be configured to simplify the computer vision problem. Usually, the number and nature of possible objects will tend to be restricted, and the visual system will be tailored to the function performed. Thus many of them are based on a pattern recognition, rather than an image understanding, approach. Industrial vision systems are characteristically used for such activities as inspection, manipulation and assembly.

A popular organization for industrial computer vision is a two-stage hierarchy with a bottom-up control flow. The lower level segments the image into regions corresponding to object surfaces. The higher level used this segmentation to identify objects from their surface descriptions.

In practice, most successful systems incorporate aspects of both bottom-up and top-down control. The bottom-up processing is used to extract prominent features of a part to determine its position. Then, top-down control is used to direct a search to determine if the part satisfies an inspection criterion.

Industrial inspection and assembly operations are well suited to model-based analysis, because of the well-defined geometric descriptions associated with manufactured items. CAD/CAM technology allows the specification of objects using either volumetric or surface-based models. These geometrically based models are particularly appropriate to the hypothesis-verify approach, in which low-level image features are extracted and matched to an appropriate computer-generated 2-D representation.

In addition to geometric models, objects may also be represented by graphs. In this case, recognition becomes a graph-matching process.

More commonly at present, rather than using geometric models or graphs, industrial vision systems are taught by being presented sample parts to be recognized in each of their expected stable states. Aspects of the resulting images are typically stored as templates, and recognition becomes template matching. The objects can also be represented in terms of their characteristic features, such as area, number of holes, etc., and the resulting feature vector stored to be matched (via a search process) to the corresponding extracted feature vector of the image during system operation.

To simplify industrial vision systems, the input is usually reduced to a binary (black and white) image, so that objects appear as silhouettes. Simplicity is important in industrial vision systems because the computation time is limited, as most systems are expected to operate in near real time.

2. Examples of Efforts in Industrial Visual Inspection Systems

Kruger and Thompson (1981) discuss some example efforts of vision systems designed for inspection. The systems reviewed are primarily for the inspection of printed circuit boards and IC chips, with template matching being the predominant inspection approach.

Chin (1982) has recently published an extensive bibliography on automated visual inspection techniques and applications.

3. Examples of Efforts in Industrial Visual Recognition and Location Systems

Table II-2 (largely derived from Kruger and Thompson, 1981) lists some example efforts of vision systems designed for industrial part recognition and location. All these systems use a bottom-up approach. It will be observed that (except for Vamos 1979, and Albus, et al., 1982) these systems utilize template or feature vector matching. Vamos does work from a 3D wire frame mode which utilizes computer graphics type techniques to transform a model projection into alignment with observed lines in the image.

Albus' Machine Vision Group in the NBS Industrial Systems Division is using simplified 3D surface models of machined parts to generate expectancy images from needed viewpoints. The group is seeking to achieve real-time, hierarchical, multi-sensory, interactive robot guidance.

4. Commercially Available Industrial Vision Systems

Gevarter (1982A) surveys many of the Industrial Vision Systems that are currently commercially available. Most of the systems require special lighting.

Many of the systems designed for verification and inspection use pattern recognition, rather than AI techniques. The systems tend to be bottom-up (see Figure II-4A) because of the speed required to achieve real-time operations. Often unique edge and feature extraction algorithms are programmed in hardware or firmware.

The more sophisticated systems tend to utilize variations and improvements on the SRI Vision Module described in Table II-2.

A few systems make good use of structured light for 3D sensing. A number of efforts in visual guidance of arc welding also utilize this technique.

I. Who Is Doing It

Rosenfeld, at the University of Maryland, issues a yearly bibliography, arranged by subject matter, related to the computer processing of pictorial information. The issue covering 1981 (Rosenfeld, 1982) includes nearly 1000 references.

The following is a list by category of the U.S. "principal players" in computer vision.

1. Research Oriented

Universities Funded Under DARPA IU Program	*Other Active Universities*
CMU	U of Texas at Austin
U of MD	VPI
MIT	Purdue
U. of Mass.	U of PA
Stanford U	U of IL
U of Rochester	Wayne State U
USC	JHU
U of Rhode Island	RPI

TABLE II-2. Example Research Efforts in Industrial Visual Recognition and Location Systems.

Developer Purpose Sample Domains	Approach	Modeling and Representations
Agin (1980) SRI Vision Module Locate, identify and guide manipulation of industrial parts Engine Parts SRI	Bottom-up approach Uses thresholding to convert to a binary image Each line is sequentially scanned and edge points (where pixels change from 1 to 0 or 0 to 1 recorded). Each resulting segment on a line is matched to the previous line to determine their overlapping relationships. Using these relationships, the program traces the appearance and disappearance of blobs (regions) as the image is processed from top to bottom. Using blob descriptors, the system can recognize parts regardless of their position or orientation. The descriptors are matched using either a binary decision tree or a normalized nearest-neighbor method. The system is trained by repeatedly showing the object to the TV camera resulting in all potentially useful shape descriptions being automatically calculated and stored	Blob descriptors include: —max. and min. x and y values —Holes —Area —Moments of inertia —Perimeter length —Linked list of coordinates on the perimeter
Holland Rossol & Ward (1979) Consight I Industrial part location, recognition and manipulation Engine parts GM	Two linear light sources superimpose a line of light on a conveyor belt perpendicular to its direction of motion. The two lines separate, proportional to the part passing by. Point of separation determines part boundary; degree of separation determines part thickness. The scene is imaged with a linear array camera and a silhouette automatically generated. Uses same feature vector approach as SRI Module.	Feature vector of part image characteristics

TABLE II-2. Example Research Efforts in Industrial Visual Recognition and Location Systems. (cont.)

Developer Purpose Sample Domains	Approach	Modeling and Representations
NBS: Albus et al. (1982) Visual servoing for robot guidance (real-time location and identification for manipulation) Machined parts	Employs a point light source, a sheets-of-structured-light generator and a camera, all mounted on the wrist of a robot arm. Uses alternate frames of: 1. A regular point source illumination of the entire object, and 2. Two parallel planes of structured light.	Uses quadratic approximations to surfaces of idealized 3-D objects.
National Bureau of Standards	System determines location and orientation based on triangulation associated with relative height of intersection of light sheets with part, and recognition based on shape and size of observed lines that the planes of light make as they intersect part. Uses this information to interpret outline seen in image produced by the point source illumination. Analysis of vision input is performed with a hierarchically organized group of microprocessors. At each level of the hierarchy, an analytic process is guided by an expectancy-generating modeling process. The modeling process is in turn driven by a store of *a priori* knowledge, by knowledge of the robot's movements, and by feedback from the analytic process. Each such level of the hierarchy provides output to guide a corresponding level of the robot's hierarchical control system.	

TABLE II-2. Example Research Efforts in Industrial Visual Recognition and Location Systems. (cont.)

Developer Purpose Sample Domains	Approach	Modeling and Representations
Perkins (1978)	Operates on 32 gray levels	Concurve models of sample parts
Industrial parts recognition	Bottom-up scene segmentation approach	
Engine components GM	1. Reduce 256×256 pixel image to an "edge gradient" image 2. Link edges with similar gradient magnitudes to form chains 3. Characterize chains as either straight lines or circular arcs. (This reduces 65,000 pixel image to about 50 concurves.)	
	System matches observed concurves with model generated concurves using:	
	1. A preset control structure to select the order in which combinations of model and scene concurves are to be matched.	
	2. Starts by matching one model and one scene concurve	
	3. The stored model is spatially transformed and rotated to fit associated scene concurves	
	System interactively trained by generating concurves of sample parts	
	Can identify parts partially occluded by other parts	

TABLE II-2. Example Research Efforts in Industrial Visual Recognition and Location Systems. (cont.)

Developer Purpose Sample Domains	Approach	Modeling and Representations
Yachida and Tsuji (1978) Industrial parts recognition Nonoccluded parts of a small gasoline engine Osaka Univ.	Uses a boundary detection and isolation of parts in a binary image approach similar to SRI Vision Module Recognition system based on a structured step-by-step analysis with the previously stored models Uses a series of special feature detectors —hole detector —line finder —texture detector —small hole detector System training involves interactive man-machine examination of the identification task	Stable orientation models of parts —part name —orientation —list of primitive features —polar coordinate boundary representation
Vamos (1979) Recognition of 3D objects Bearing housings Assembly Sheet metal parts to be painted Neural nets in microscopic-section in neural research Hungarian Acad. of Science	Finds edges using a simplified version of the Hueckel-operator using only two linear templates Lines are then fitted to edges Wire-frame model transformed (and hidden line elimination used) to correspond to image—yielding recognition and part orientation Objects are interactively taught to system either by building a geometric model or by a computer-aided transformation of viewed samples	3D Wire Frame Models

Non-Profits
SRI International, AI Center
JPL
ERIM

U.S. Government
NBS, Industrial Systems Div., Gaithersburg, MD
NOSC (Naval Ocean Systems Center), San Diego
NIH (National Institutes of Health)

2. Commercial Vision Systems Developers

Hundreds of companies are now involved in vision systems, a partial listing being given in Gevarter (1982A).

J. Summary of the State-of-the-Art

1. Human Vision

Human vision is the only available example of a general purpose vision system. However, thus far not many AI researchers have taken an interest in the computations performed by natural visual systems, but this situation is changing.

The MIT vision group (among others) believes that, to a first approximation, the human visual system is subdivided into modules specializing in visual tasks. There is also evidence that people do global processing first and use it to constrain local processing.

Considerable information now exists about lower level visual processing in humans. However, as we progress up the human visual computing hierarchy, the exact nature of the appropriate representations becomes subject to dispute. Thus, overall human visual perception is still very far from being understood.

2. Low and Intermediate Levels of Processing

Though methods for powerful high-level understanding visual analysis are still in the process of being determined, insights into low-level vision are emerging. The basic physics of imaging, and the nature of constraints in vision and their use in computation is fairly well understood. Detailed programs for vision modules, such as "shape from shading" and "optical flow," have begun to appear. Also, the representational issues are now better understood.

However, even for well understood low-level operations such as edge detection, (see, e.g., Ballard, 1982) there has been no convergence among the many techniques proposed, and no method stands out as the best. In general, edge detectors are still unreliable, though Marr and Hilbert's approach, based on the zero crossing of the second derivative of the intensity gradient, appears promising.

In industrial vision, the primary technique for achieving robust edge finding and segmentation is to use special lighting and convert to a silhouette binary image in which edges and regions are readily distinguishable.

At intermediate levels, edge classification and labelling have been very successfully used in the blocks world.

Binford (1982) in reviewing existing research in model-based vision systems observed that most systems first segment regions then describe their shape. None of the systems makes effective use of texture for segmentation and description. In general, shape description is primitive and interpretation systems have not yet made full use of even these limited capabilities.

As yet, the extraction of useful information from color is extremely rudimentary. The perceptual use of motion (optical flow) has been a focus of attention recently, but findings are preliminary.

For low level processing, many recent algorithms take the form of parallel computations involving local interactions. One popular approach having this character is "relaxation," in which local computations are iteratively propagated to try to extract global features. These locally parallel architectures are well suited to rapid parallel processing techniques using special purpose VLSI chips.

3. Industrial Vision Systems

Barrow and Tenenbaum (1981, p. 572) observe that:

> Significant progress has been made in recent years on practical applications of machine vision. Systems have been developed that achieve useful levels of performance on complex real imagery in tasks such as inspection of industrial parts, interpretation of aerial imagery, and analysis of chest x-rays. Virtually all such systems are special purpose, being heavily dependent on domain-specific constraints and techniques.

It has been estimated that as of mid-1982, though less than 50 sophisticated industrial vision systems were actually in use in the U.S., approximately 1000 simple line-scan inspection systems were in regular operation. Though special purpose systems have thus far been the most effective, successful vision applications are now becoming commonplace and are expanding. Vision manufacturers are now beginning to provide easier user programming, friendlier user interfaces, and systems engineering support to prospective users. Many firms are now entering the industrial vision field, with technical leap-frogging being common due to rapidly changing technology.

4. General Purpose Vision Systems

Though many practical image recognition systems have been developed, Hiatt (1981, pp. 2, 8) observes that, "In current vision applications, the type of scene to be processed and acted upon is usually carefully defined and limited to the capability of the machine . . . General purpose computer vision has not yet been solved in practice." This domain specificity makes each new application expensive and time consuming to develop.

Binford (1982) in reviewing current model-based research vision systems concludes that most systems have not attempted to be general vision systems, though ACRONYM does demonstrate some progress toward this goal. Existing vision systems performances are strongly limited by the performance of their segmentation modules, their weak use of world knowledge and weak descriptions, making little use of shape.

With the exception of ACRONYM (and to an extent 3-D Mosaic), the systems surveyed depend on image models and relations, and therefore are strongly viewpoint-dependent. To generalize to viewpoint-insensitive interpretations would require three-dimensional modeling and interpretation as in ACRONYM.

Binford concludes that though the results of these and other efforts are encouraging as first demonstrations, nevertheless as general vision systems, they have a long way to go.

K. Applications and Future Trends

Brady (1981, p. 2) states that, "There is currently a surge of interest in image understanding on the part of industry." Examples of current computer vision applications are indicated in Figure II-6.

As the field of computer vision unfolds, we expect to see the following future trends.*

1. Techniques

- Though most industrial vision systems have used binary representations, we can expect increased use of gray scales because of their potential for handling scenes with cluttered backgrounds and uncontrolled lighting.
- Recent theoretical work on monocular shape interpretation from images (shape from shading, texture, etc.) make it appear promising that general mechanisms for generating spatial observations from images will be available within the next 2 to 5 years to support general vision systems.
- Successful techniques (such as stereo and motion parallax) for deriving shape and/or motion from multiple images should also be available within 2 to 5 years.
- The mathematics of Image Understanding will continue to become more sophisticated.
- Enlargement will continue of the links now growing between Image Understanding and Theories of Human Vision.

2. Hardware and Architecture

- We are now seeing hardware and software emerging that enables real-time operation in simple situations. Within the next 2 to 5 years we should see hardware and software that will enable similar real-time operation for robotics and other activities requiring recognition, and position and orientation information.
- Fast raster-based pipeline preprocessing hardware to compute low-level features in local regions of an entire scene are now becoming available and should find general use in commercial vision systems in 2 to 4 years.
- As at virtually all visual levels, processing seems inherently parallel, parallel processing is a wave of the future (but not the entire answer).
- Relaxation and constraint analysis techniques are on the increase and will be increasingly reflected in future architectures.

3. AI and General Vision Systems

Computer vision will be a key factor in achieving many artificial intelligence applications. The goal is to move from special-purpose visual processing to general-purpose computer vision. Work to date in model-based systems has made a tentative beginning. But the long-run goal is to be able

*These trends have been largely derived from statements by Brady (1981A, 1981B), Binford (1982), Kruger and Thompson (1981), Agin (1980), Arden (1980), Rosenfeld (1981), Hiatt (1981), and Barrow and Tenenbaum (1981).

AUTOMATION OF INDUSTRIAL PROCESSES

Object acquisition by robot arms, for example, for sorting or packing items arriving on conveyor belts.
Automatic guidance of seam welders and cutting tools.
VLSI-related processes, such as lead bonding, chip alignment and packaging.
Monitoring, filtering, and thereby containing the flood of data from oil drill sites or from seismographs.
Providing visual feedback for automatic assembly and repair.

INSPECTION TASKS

The inspection of printed circuit boards for spurs, shorts, and bad connections.
Checking the results of casting processes for impurities and fractures.
Screening medical images such as chromosome slides, cancer smears, x-ray and ultrasound images, tomography.
Routine screening of plant samples.
Inspection of alpha-numerics on labels and manufactured items.
Checking packaging and contents in pharmaceutical and food industries.
Inspection of glass items for cracks, bubbles, etc.

REMOTE SENSING

Cartography, the automatic generation of hill-shaded maps, and the registration of satellite images with terrain maps.
Monitoring traffic along roads, docks, and at airfields.
Management of land resources such as water, forestry, soil erosion, and crop growth.
Detecting mineral ore deposits.

MAKING COMPUTER POWER MORE ACCESSIBLE

Management information systems that have a communication channel considerably wider than current systems that are addressed by typing or pointing.
Document readers (for those who still use paper).
Design aids for architects and mechanical engineers.

MILITARY APPLICATIONS

Tracking moving objects.
Automatic navigation based on passive sensing.
Target acquisition and range finding.

AIDS FOR THE PARTIALLY SIGHTED

Systems that read a document and speak what they read.
Automatic "guide dog" navigation systems.

Figure II-6. Examples of Applications of Computer Vision Now Underway.

to deal with unfamiliar or unexpected input.* Reasoning in terms of generic models and reasoning by analogy are two approaches being pursued. However, it is anticipated that it will be a decade or more before substantial progress will be made.

4. Modeling and Programming

- Now emerging is 3D modeling, arising largely from CAD/CAM technology. 3D CAD/CAM data bases will be integrated with industrial vision systems to realistically generate synthesized images for matching with visual inputs.
- Illumination models, shading and surface property models will be increasingly incorporated into visual systems.
- Volumetric models which allow prediction and interpretation at the levels of volumes, rather than images, will see greater utilization.
- High level vision programming languages (such as Automatix's RAIL) that can be integrated with robot and industrial manufacturing languages are now beginning to appear and will become commonplace within 5 years.
- Generic representations for amorphous objects (such as trees) have been experimentally utilized and should become generally available within 5 years.

5. Knowledge Acquisition

- Strategies for indexing into a large database of models should be available within the next 2 to 5 years.
- "Training by being told" will supplement "training by example" as computer graphics techniques and vision programming languages become more common.

6. Sensing

- An important area of development is 3D sensing. Several current industrial vision systems are already employing structured light for 3D sensing. A number of new innovative techniques in this area are expected to appear in the next 5 years.
- More active vision sensors such as lidar are now being explored, but are unlikely to find substantial industrial application until the last half of this decade.

7. Industrial Vision Systems

- We will see increased use of advanced vision techniques in industrial vision systems, including gray scale imagery.
- We are now observing a shortening time lag between research advances and their applications in industry. It is anticipated that in the future this lag may be as little as one to two years.
- Advanced electronics hardware at reduced cost is increasing the capabilities and speed of industrial vision, while simultaneously reducing costs.

---

*As computer vision systems move toward this goal, they will increasingly incorporate Expert System components using multiple knowledge sources. Gevarter (1982B) provides An Overview of Expert Systems, in which ACRONYM and VISIONS are considered to be examples of Expert Systems.

- It is anticipated that special lighting and active sensing will play an increasing role in industrial vision.
- Common programming languages and improved interface standards will within the next 3 to 10 years enable easier integration of vision to robots and into the industrial environment.

8. Future Applications

- It is anticipated that about one quarter of all industrial robots will be equipped with some form of vision system by 1990.
- It is likely that in the order of 90% of all industrial inspection activities requiring vision will be done with computer vision systems within the next decade.
- New vision system applications in a wide variety of areas, as yet unexplored, will begin to appear within this decade. An example of such a system might be visual traffic monitors at intersections that could perceive cars, pedestrians, etc., in motion, and control the flow of traffic accordingly.
- Computer vision will play a large role in future military applications. The Defense Mapping Agency intends to achieve fully automated production for mapping, charting and geodesy by 1995, utilizing "expert system"-guided computer vision facilities.

L. Conclusion

In conclusion, the amount of activity and the many researchers in the computer vision field suggest that within the next 5 to 10 years, we should see some startling advances in practical computer vision, though the availability of practical general vision systems still remains a long way off.

REFERENCES

- Albus, J., Kent, E., Nashman, M., Mansbach, P., and Palombo, L., "A 6-D Vision System," *Proceedings of SPIE Technical Symposium East '82,* Arlington, VA, May 3-7, 1982.
- Agin, G. J., "Computer Vision Systems for Industrial Inspection and Assembly," *Computer,* May 1980.
- Arden, B. W., (eds.), *What Can Be Automated?* Cambridge: M.I.T., 1980, pp. 482-487.
- Ballard, D. H. and Brown, C. M., *Computer Vision,* Englewood Cliffs: Prentice Hall, 1982
- Barrow, H. G. and Tenenbaum, J. M., "Computational Vision," *Proceedings of the IEEE,* Vol. 69, No. 5, May 1981, pp. 572-595.
- Binford, T. O., "Survey of Model-based Image Analysis Systems," *Robotics Research,* Vol. 1, No. 1, Spring 1982.
- Brady, M., "Computational Approaches to Image Understanding," M.I.T. A.I. Memo No. 653, October 1981A. (Also in Computing Survey, Vol. 14, No. 1, Mar. 1982, pp. 3-71).
- Brady, M., "The Changing Shape of Computer Vision,"*Artificial Intelligence,* 17 (1-3), 1981B, pp. 1-15.
- Brooks, R., Greiner, R. and Binford, T.O., The ACRONYM Model-Based Vision System, *Proc. Int. Jt, Conf. Artificial Intelligence 1979,* 6, 105-113.
- Brooks, R., "Symbolic Reasoning Among 3-D Models and 2-D Images," *AI* 17, 1981, pp. 285-348.
- Chin, R. T., "Automated Visual Inspection Techniques and Applications: A Bibliography," Pattern Recognition, Vol. 15 (4), 1982, pp. 343-357.

• Cohen, P. R., and Feigenbaum, E. A., Chap. XIII, Vision, *The Handbook of Artificial Intelligence,* Vol. III, Los Altos, CA: Kaufmann, 1982, pp. 125-321.

• Gevarter, W. B., "A Wiring Diagram of the Human Brain as a Model for Artificial Intelligence," *Proc. of the IEEE Inter. Conf. on Cyb. and Society,* Wash., D.C., Sept. 1977, pp. 694-698.

• Gevarter, W. B., *An Overview of Computer Vision,* NSIR 82-2582, National Bureau of Standards, Wash., D.C., Sept. 1982A.

• Gevarter, W. B., *An Overview of Expert Systems,* NBSIR 82-2505, National Bureau of Standards, Wash., D.C., May 1982B.

• Hanson, A. R. and Riseman, E. M., (eds) *Computer Vision Systems,* New York: Academic Press, 1978a.

• Hanson, A. R. and Riseman, E. M. (1978b), "Segmentation of Natural Scenes," in Hanson and Riseman (1978a) pp. 129-163.

• Hiatt, B., "Toward Machines that See," *Mosaic,* Nov/Dec 1981, pp. 2-8.

• Holland, S. W., Rossol, L. and Ward, M. R., "Consight-I: A Vision Controlled Robot System for Transferring Parts from Belt Conveyors," *Computer Vision and Sensor-Based Robots,* G. G. Dodd, and L. Rossol, Eds. New York: Plenum Press, 1979, pp. 81-100.

• Kanade, T., "Model Representation and Control Structures in Image Understanding," *Proc. of IJCAI-77,* Cambridge, Aug. 1977, 1074-1082.

• Kashioka, S. Fjiri, M., and Sakamoto, Y., "A Transistor Wire Bonding System Utilizing Multiple Local Pattern Machine," *IEEE Trans. Syst. Mar. Cybern.,* vol. SMC-6 1976.

• Kruger, R. P. and Thompson, W. B., "A Technical and Economic Assessment of Computer Vision for Industrial Inspection and Robotic Assembly," *Proceedings of the IEEE,* Vol. 69, No. 12, Dec. 1981, pp. 1524-1538.

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

• Marr, D., and Nishihara, H., "Visual Information Processing: Artificial Intelligence and the Sensorium of Sight," *Technology Review,* October 1978, pp. 28-47.

• Minsky, M. L., "A Framework for Representing Knowledge," In P. H. Winston (Ed), *The Psychology of Computer Vision,* New York: McGraw-Hill, 1975, pp. 211-277.

• Parma, C. C., Hanson, A. M., Riseman, E. M., "Experiments in Schema-Driven Interpretation of a Natural Scene," U. of Mass., COINS Tech. Rept 80–10, 1980.

• Perkins, W. A. "Model-Based Vision System for Industrial Parts," *IEEE, Trans. Comput.,* vol. C-27, 1978.

• Pratt, W. K., *Digital Image Processing,* New York: Wiley, 1978, pp. 568-587.

• Rosenfeld, A. "Picture Processing, 1981," Computer Vision Lab Rept. TR-1134, U of MD, College Park, Jan 1982. (Also in *CGIP,* May 1982).

• Tenenbaum, et al., "Prospects for Industrial Vision," in *Computer Vision and Sensor-Based Robots,"* G. G. Dodd and L. Rossol (Eds.), NY: Plenum Press, 1979, pp. 239-259.

• Vamos, T., Bathor, M., and Mero, L., "A Knowledge-Based Interactive Robot-Vision System," *6IJCAI,* 1979, pp. 920-922.

• Yachida, M. and Tsuji, S., "A Versatile Machine Vision System for Complex Industrial Parts," *IEEE Trans. Comput.,* vol. C-26, 1977.

• DARPA Image Understanding Workshops (Annual).

• IEEE Computer Society Conferences on Computer Vision and Pattern Recognition (Annual).

III. NATURAL LANGUAGE PROCESSING (NLP)*

A. Introduction

One major goal of Artificial Intelligence (AI) research has been to develop the means to interact with machines in natural language (in contrast to a computer language). The interaction may be typed, printed or spoken. The complementary goal has been to understand how humans communicate. The scientific endeavor aimed at achieving these goals has been referred to as computational linguistics (or more broadly as cognitive science), an effort at the intersection of AI, linguistics, philosophy and psychology.

Human communication in natural language is an activity of the whole intellect. AI researchers, in trying to formalize what is required to properly address natural language, find themselves involved in the long term endeavor of having to come to grips with this whole activity. (Formal linguists tend to restrict themselves to the structure of language.) The current AI approach is to conceptualize language as a knowledge-based system for processing communications and to create computer programs to model that process.

Communication acts can serve many purposes, depending on the goals, intentions and strategies of the communicator. One goal of the communication is to change some aspect of the recipient's mental state. Thus, communication endeavors to add or modify knowledge, change a mood, elicit a response or establish a new goal for the recipients.

For a computer program to interpret a relatively unrestricted natural language communication, a great deal of knowledge is required. Knowledge is needed of:

— the structure of sentences
— the meaning of words
— the morphology of words
— a model of the beliefs of the sender
— the rules of conversation, and
— an extensive shared body of general information about the world.

This body of knowledge can enable a computer (like a human) to use expectation-driven processing in which knowledge about the usual properties of known objects, concepts, and what typically happens in situations, can be used to understand incomplete or ungrammatical sentences in appropriate contexts.

B. Applications

There are many applications for computer-based natural language understanding systems. Some of these are listed in Table III-1.

*A more complete treatment of NLP is given in Gevarter (1983).

TABLE III-1. Some Applications of Natural Language Processing.

Discourse	**Interaction with Intelligent Programs**
Speech Understanding	Expert Systems Interfaces
Story Understanding	Decision Support Systems
Information Access	Explanation Modules for Computer Actions
	Interactive Interfaces to Computer Programs
Information Retrieval	**Interacting with Machines**
Question Answering Systems	Control of Complex Machines
Computer-Aided Instruction	
Information Acquisition or Transformation	**Language Generation**
Machine Translation	Document or Text Generation
Document or Text Understanding	Speech Output
Automatic Paraphrasing	Writing Aids: e.g., grammar checking
Knowledge Compilation	
Knowledge Acquisition	

C. Approach

Natural Language Processing (NLP) systems utilize both linguistic knowledge and domain knowledge to interpret the input. As domain knowledge (knowledge about the subject area of communication) is so important to understanding, it is usual to classify the various systems based on their representation and utilization of domain knowledge. On this basis, Hendrix and Sacerdoti (1981) classify systems as Types A, B, or C,* with Type A being the simplest, least capable and correspondingly least costly systems.

1. Type A: No World Models

a. Key Words or Patterns

The simplest systems utilize ad hoc data structures to store facts about a limited domain. Input sentences are scanned by the programs for predeclared key words, or patterns, that indicate known objects or relationships.

b. Limited Logic Systems

In limited logic systems, information in their data base was stored in some formal notation, and language mechanisms were utilized to translate the input into the internal form. The internal form chosen was such as to facilitate performing logical inferences on information in the data base.

*Other system classifications are possible, e.g., those based on the range of syntactic coverage.

2. Type B: Systems That Use Explicit World Models

In these systems, knowledge about the domain is explicitly encoded, usually in frame or network representations (discussed in a later section) that allow the system to understand input in terms of context and expectations. Cullingford's work (see Schank and Ableson, 1977) on SAM (Script Applier Mechanism) is a good example of this approach.

3. Type C: Systems that Include Information about the Goals and Beliefs of Intelligent Entities.

These advanced systems (still in the research stage) attempt to include in their knowledge base information about the beliefs and intentions of the participants in the communication. If the goal of the communication is known, it is much easier to interpret the message. Schank and Abelson's (1977) work on plans and themes reflects this approach.

D. The Parsing Problem

For more complex systems than those based on key words and pattern matching, language knowledge is required to interpret the sentences. The system usually begins by "parsing" the input (processing an input sentence to produce a more useful representation for further analysis). This representation is normally a structural description of the sentence indicating the relationship of the component aparts. To address the parsing problem and to interpret the result, the computational linguistic community has studied syntax, semantics, and pragmatics. Syntax is the study of the structure of phrases and sentences. Semantics is the study of meaning. Pragmatics is the study of the use of language in context.

E. Grammars

Barr and Feigenbaum (1981, p. 229) state, "A grammar of a language is a scheme for specifying the sentences allowed in the language, indicating the syntactic rules for combining words into well-formed phrases and clauses." The following grammars are some of the most important.*

1. Phrase Structure Grammar — Context Free Grammar

Chomsky (see, e.g., Winograd, 1983) had a major impact on linguistic research by devising a mathematical approach to language. He defined a series of grammars based on rules for rewriting sentences into their component parts. He designated these as 0, 1, 2, or 3, based on the restrictions associated with the rewrite rules, with 3 being the most restrictive.

Type 2 — Context-Free (CF) or Phrase Structure Grammar (PSG) — has been one of the most useful in natural-language processing. It has the advantage that all sentence structure derivations can be represented as a tree and practical parsing algorithms exist. Though it is a relatively natural grammar, it is unable to capture all the sentence constructions found in most natural languages such as English. Gazder (1981) has recently broadened the applicability of CF PSG by adding augmentations to handle situations that do not fit the basic grammar. This generalized Phrase Structure Grammar is now being developed by Hewlett Packard (Gawron et al., 1982).

---

*Charniak and Wilks (1976) provide a good overview of the various approaches.

2. Transformational Grammar

Tennant (1981, p. 89) observes that "The goal of a language analysis program is recognizing grammatical sentences and representing them in a canonical structure (the underlying structure)." A transformational grammar (Chomsky, 1957) consists of a dictionary, a phrase structure grammar and a set of transformations. In analyzing sentences, using a phrase structure grammar, first a parse tree is produced. This is called the surface structure. The transformational rules are then applied to the parse tree to transform it into a canonical form called the deep (or underlying) structure. As the same thing can be stated in several different ways, there may be many surface structures that translate into a single deep structure.

3. Case Grammar

Case Grammar is a form of Transformational Grammar in which the deep structure is based on cases - semantically relevant syntactic relationships. The central idea is that the deep structure of a simple sentence consists of a verb and one or more noun phrases associated with the verb in a particular relationship. These semantically relevant relationships are called cases. Fillmore (1971) proposed the following cases: Agent, Experiencer, Instrument, Object, Source, Goal, Location, Type and Path.

The cases for each verb form an ordered set referred to as a "case frame." A case frame for the verb "open" would be:

(object (instrument) (agent))

which indicates that open always has an object, but the instrument or agent can be omitted as indicated by their surrounding parentheses. Thus the case frame associated with the verb provides a template which aids in understanding a sentence.

4. Semantic Grammars

For practical systems in limited domains, it is often more useful, instead of using conventional syntactic constituents such as noun phrases, verb phrases and prepositions, to use meaningful semantic components instead. Thus, in place of nouns when dealing with a naval data base, one might use ships, captains, ports and cargos. This approach gives direct access to the semantics of a sentence and substantially simplifies and shortens the processing. Grammars based on this approach are referred to as semantic grammars (see, e.g., Burton, 1976).

5. Other Grammars

A variety of other, but less prominent, grammars have been devised. Still others can be expected to be devised in the future. One example is Montague Grammar (Dowty et al., 1981) which uses a logical functional representation for the grammar and therefore is well suited for the parallel-processing logical approach now being pursued by the Japanese (see Nishida and Doshita, 1982) for their future AI work as embodied is their Fifth Generation Computer research project.

F. Semantics and the Cantankerous Aspects of Language

Semantic processing (as it tries to interpret phrases and sentences) attaches meanings to the words. Unfortunately, English does not make this as simple as looking up the word in the dictionary, but provides many difficulties which require context and other knowledge to resolve. Examples are:

1. Multiple Word Senses

Syntactic analysis can resolve whether a word is used as a noun or a verb, but further analysis is required to select the sense (meaning) of the noun or verb that is actually used. For example, "fly" used as a noun may be a winged insect, a fancy fishhook, a baseball hit high in the air, or several other interpretations as well. The appropriate sense can be determined by context (e.g., for "fly" the appropriate domain of interest could be extermination, fishing or sports), or by matching each noun sense with the senses of other words in the sentence. This latter approach was taken by Reiger and Small (1979) using the (still embryonic) technique of "interacting word experts," and by Finin (1980) and McDonald (1982) as the basis for understanding noun compounds.

2. Pronouns

Pronouns allow a simplified reference to previously used (or implied) nouns, sets or events. Where feasible, using pragmatics, pronoun antecedents are usually identified by reference to the most recent noun phrase having the same context as the pronoun.

3. Ellipsis and Substitution

Ellipsis is the phenomenon of not stating explicitly some words in a sentence, but leaving it to the reader or listener to fill them in. Substitution is similar — using a dummy word in place of the omitted words. Employing pragmatics, ellipses and substitutions are usually resolved by matching the incomplete statement to the structures of previous recent sentences — finding the best partial match and then filling in the rest from this matching previous structure.

G. Knowledge Representation*

As the AI approach to natural language processing is heavily knowledge based, it is not surprising that a variety of knowledge representation (KR) techniques have found their way into the field. Some of the more important ones are:

1. Procedural Representations — The meanings of words or sentences being expressed as computer programs that reason about their meaning.

*More complete presentations on KR can be found in Chapter III of Barr and Feigenbaum (1981), and in Part C of this volume.

2. *Declarative Representations*

a. Logic — Representation in First Order Predicate Logic, for example.

b. Semantic Networks — Representations of concepts and relationships between concepts as graph structures consisting of nodes and labeled connecting arcs.

3. *Case Frames* — (covered earlier)

4. *Conceptual Dependency* — This approach (related to case frames) is an attempt to provide a representation of all actions in terms of a small number of semantic primitives into which input sentences are mapped (see, e.g., Schank and Riesbeck, 1981). The system relies on 11 primitive physical, instrumental and mental ACT's (propel, grasp, speak, attend, P trans, A trans, etc.), plus several other categories or concept types.

5. *Frame* — A complex data structure for representing a whole situation, complex object or series of events. A frame has slots for objects and relations appropriate to the situation.

6. *Scripts* — Frame-like data structures for representing stereotyped sequences of events to aid in understanding simple stories.

H. Syntactic Parsing

Parsing assigns structures to sentences. The following types have been developed over the years for NLP. (Barr and Feigenbaum, 1981).

1. *Template Matching:* Most of the early (and some current) NL programs performed parsing by matching their input sentences against a series of stored templates.

2. *Transition Nets:*

Phrase structure grammars can be syntactically decomposed using a set of rewrite rules such as indicated in Figure III-1. Observe that a simple sentence can be rewritten as a Noun Phrase and a Verb Phrase as indicated by:

$$S \rightarrow NP\ VP$$

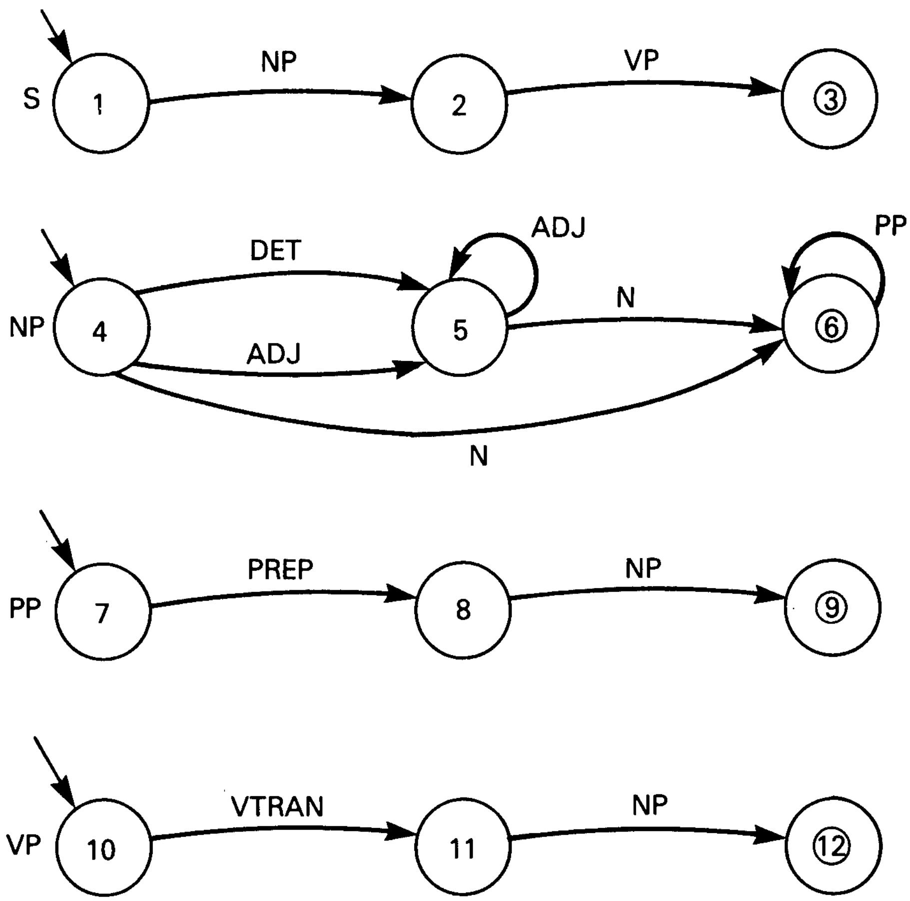
The noun phrase can be rewritten by the rule

$$NP \rightarrow (DET)(ADJ^*)N(PP^*)$$

where the parentheses indicate that the item is optional, while the asterisks (associated with the adjectives and prepositional phrases) indicate that any number of items may occur.

An example of an analyzed noun phrase is shown in Figures III-2 and III-3.

As the transition networks analyze a sentence, they can collect information about the word patterns they recognize and fill slots in a frame associated with each pattern. Thus, they can identify noun phrases as singular or plural, whether the nouns refer to persons and if so their gender, etc., needed to produce a deep structure. A simple approach to collecting this information is to attach subroutines to be called for each transition. A transition network with such subroutines attached is called an "augmented transition network," or ATN. With ATN's, word patterns can be



GRAMMAR

S ⟶ NP VP
NP ⟶ (DET) (ADJ*) N (PP*)
PP ⟶ PREP NP
VP ⟶ VTRAN NP

Figure III-1. A Transition Network for a Small Subset of English.

Each diagram represents a rule for finding the corresponding word pattern. Each rule can call on other rules to find needed patterns.

After Graham (1979, p214.)

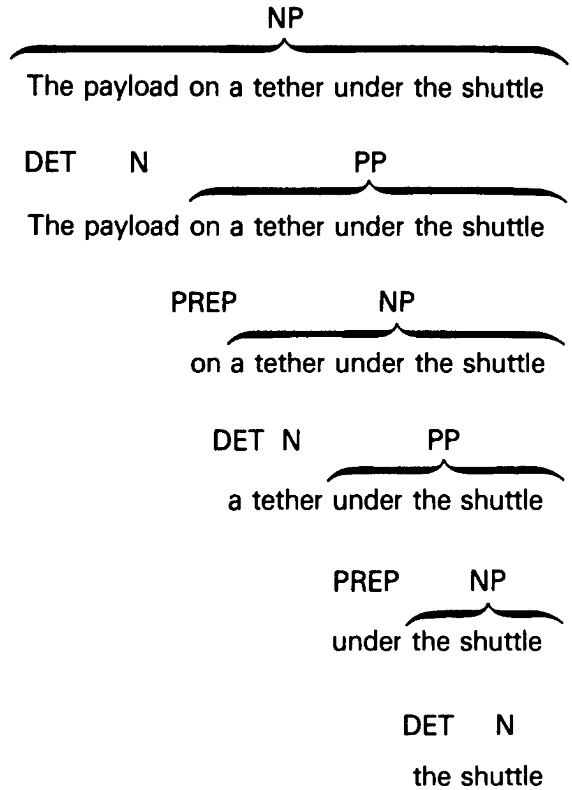


Figure III-2. Example Noun Phrase Decomposition.

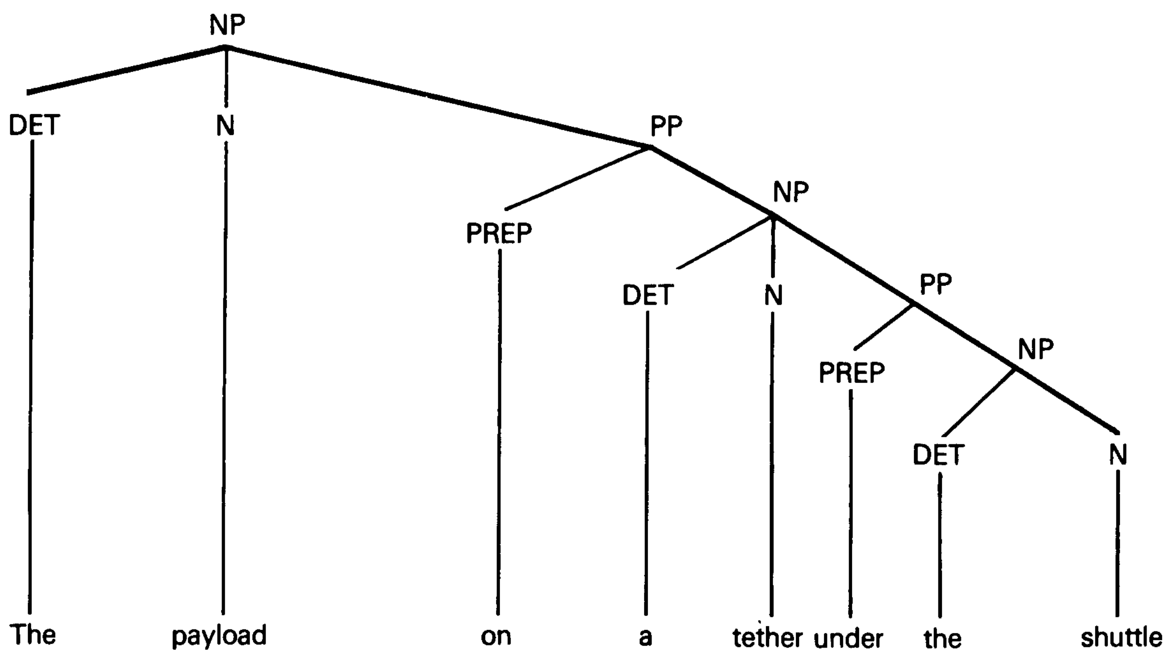


Figure III-3. Parse Tree Representation of the Noun Phrase Surface Structure.

recognized. For each word pattern, we can fill slots in a frame. The resulting filled frames provide a basis for further processing.

3. Other Parsers

Other parsing approaches have been devised, but ATN's remain the most popular syntactic parsers. ATN's are top-down parsers in that the parsing is directed by an anticipated sentence structure. An alternative approach is bottom-up parsing, which examines the input words along the string from left to right, building up all possible structures to the left of the current word as the parser advances. A bottom-up parser could thus build many partial sentence structures that are never used, but the diversity could be an advantage in trying to interpret input word strings that are not clearly delineated sentences or contain ungrammatical constructions or unknown words. There have been recent attempts to combine the top-down with the bottom-up approach for NLP in a similar manner as has been done for Computer Vision.

For a recent overview of parsing approaches see Slocum (1981).

I. Semantics, Parsing and Understanding

The role of syntactic parsing is to construct a parse tree or similar structure of the sentence to indicate the grammatical use of the words and how they are related to each other. The role of the semantic processing is to establish the meaning of the sentence. This requires facing up to all the cantankerous ambiguities discussed earlier.

Charniak (1981) observes that there have been two main lines of attack on word sense ambiguity. One is the use of discrimination nets (Reiger and Small, 1979) that utilize the syntactic parse tree (by observing the grammatical role that the word plays, such as taking a direct object, etc.) in helping to decide the word sense. The other approach is based on the frame/script idea (used, e.g., for story comprehension) that provides a context and the expected sense of the word (see e.g., Schank and Abelson, 1977).

Charniak indicates that the semantics at the level of the word sense is not the end of the parsing process, but what is desired is understanding or comprehension (associated with pragmatics). Here the use of frames, scripts and more advanced topics such as plans, goals, and knowledge structures (see, e.g. Schank and Riesbeck, 1981) play an important role.

J. Natural Language Processing (NLP) Systems

As indicated below, various NLP systems have been developed for a variety of functions.

1. Kinds

a. Question Answering Systems

Question answering natural language systems have perhaps been the most popular of the NLP research systems. They have the advantage that they usually utilize a data-base for a limited domain and that most of the user discourse is limited to questions.

b. Natural Language Interfaces (NLI's)

These systems are designed to provide a painless means of communicating questions or instructions to a complex computer program.

c. Computer-Aided Instruction (CAI)

Arden (1980, p. 465) states:

> One type of interaction that calls for ability in natural languages is the interaction needed for effective teaching machines. Advocates of computer-aided instruction have embraced numerous schemes for putting the computer to use directly in the educational process. It has long been recognized that the ultimate effectiveness of teaching machines is linked to the amount of intelligence embodied in the programs. That is, a more intelligent program would be better able to formulate the questions and presentations that are most appropriate at a given point in a teaching dialogue, and it would be better equipped to understand a student's response, even to analyze and model the knowledge of the student, in order to tailor the teaching to his needs.

d. Discourse

Systems that are designed to understand discourse (extended dialogue) usually employ pragmatics. Pragmatic analysis requires a model of the mutual beliefs and knowledge held by the speaker and listener.

e. Text Understanding

Though Schank (see Schank and Riesbeck, 1981) and others have addressed themselves to this problem, much more remains to be done. Techniques for understanding printed text include scripts and causative approaches.

f. Text Generation

There are two major aspects of text generation: one is the determination of the content and textual shape of the message, the second is transforming it into natural language. There are two approaches for accomplishing this. The first is indexing into canned text and combining it as appropriate. The second is generating the text from basic considerations. McDonald's thesis (1980) provides one of the most sophisticated approaches to text generation.

2. Research NLP Systems

Until recently, virtually all of the NLP systems generated were of a research nature. These NLP systems basically were aimed at serving five functions:

a. Interfaces to Computer Programs
b. Data Base Retrieval
c. Text Understanding
d. Text Generation
e. Machine Translation

Gevarter (1983) includes a survey of research NLP systems.

3. Commercial Systems:

The commercial systems available today (together with their approximate price) are listed in Table III-2. Several of these systems are derivatives of past research NLP systems.

TABLE III-2. Some Commercial Natural Language Systems.

System	Organization	Purpose	Comments
INTELLECT (Derivative of ROBOT) $50K/System (also distributed as ON-LINE ENGLISH and GRS Executive)	Artificial Intelligence Corp. Waltham, Mass (Culliane) (Information Sciences)	NLI for Data Base Retrieval (Other extensions underway)	• Several hundred systems sold • Takes about 2 weeks to implement for a new data base. • Written in PL-1 • Available for mainframes
PEARL (Based on SAM and PAM) $250K/system	Cognitive Systems New Haven, Conn	Custom NLI's The first system—Explorer—is an interface to an existing map generating system. Others are interfaces to data bases.	• Large start-up cost in building the knowledge base. • Several systems have been, and are being, built. • Written in LISP
Straight Talk (Derivative of LIFER) $660	Dictaphone, Written by Symantec Sunnyvale, CA	Highly portable NLI for DBMS for micro-computers.	• Written in PASCAL. Designed to be very compact and efficient. Available about Nov. 1983. • User customized.
SAVVY $950	SAVVY Marketing International Sunnyvale, CA	System Interface for micro-computers	• Not linguistic. Uses adaptive (best fit) pattern matching to strings of characters. • Released 3/82 • User customized
Weidner System $16K/language direction	Weidner Communications Corp. Provo, UT	Semi-Automatic Natural Language Translation.	• Linguistic approach. Written in FORTRAN IV. • Translation with human editing is approximately 100 words/hr (up to eight times as fast as human alone). • Approx. 20 sold by end of 1982, mainly to large multi-national corporations.
ALPS	ALPS Provo, UT	Interactive Natural Language Translation	• Linguistic Approach • Uses a dictionary that provides the various translations for technical words as a display to human translator, who then selects among the displayed words.

TABLE III-2. Some Commercial Natural Language Systems (cont.)

System	Organization	Purpose	Comments
NLMENU	Texas Instruments, Inc. Dallas, TX	NLI to Relational Data Bases	• Menu Driven NL Query System • All queries constructed from menu fall within linguistic and conceptual coverage of the system. Therefore, all queries entered are successful. • Grammars used are semantic grammars written in a context-free grammar formalism. • Producing an interface to any arbitrary set of relations is automated and only requires a 15-30 minute interaction with someone knowledgeable about the relations in question. • System will be available late in 1983 as a software package for a micro-computer.

K. State of the Art

It is now feasible to use computers to deal with natural language input in highly restricted contexts. However, interacting with people in a facile manner is still far off, requiring understanding of where people are coming from — their knowledge, goals and moods.

In today's computing environment, the only systems that perform robustly and efficiently are Type A systems — those that do not use explicit world models, but depend on key word or pattern matching and/or semantic grammars. In actual working systems, both understanding and text generation, ATN-like grammars can be considered the state of the art.

L. Principal U.S. Participants in NLP

*1. Research and Development**

Non-Profit

SRI
MITRE

*A review of current research in NLP is given in Kaplan (1982).

Universities

Yale U. — Dept. of Computer Science
U. of CA, Berkeley — Computer Science Div., Dept. of EECS.
Carnegie-Mellon U. — Dept. of Computer Science
U. of Illinois, Urbana — Coordinated Science Lab.
Brown U. — Dept. of Computer Science
Stanford U. — Computer Science Dept.
U. of Rochester — Computer Science Dept.
U. of Mass., Amherst — Department of Computer and Information Science
SUNY, Stoneybrook, Dept. of Computer Science
U. of CA, Irvine, Computer Science Dept.
U of PA — Dept. of Computer and Infor. Science
GA Institute of Technology — School of Infor. and Computer Science
USC — Infor. Science Institute.
MIT — AI Lab.
NYU — Computer Science Dept. and Linguistic String Project
U. of Texas at Austin — Dept. of Computer Science
Cal. Inst. of Tech.
Brigham Young U. — Linguistics Dept.
Duke U. — Dept. of Computer Science
N. Carolina State — Dept. of Computer Science
Oregon State U. — Dept. of Computer Science
Purdue U.

Industrial

BBN
TRW Defense Systems
IBM, Yorktown Heights, N.Y.
Burroughs
Sperry Univac
Systems Development Corp., Santa Monica
Hewlett Packard
Martin Marietta, Denver
Texas Instruments, Dallas
Xerox PARC
Bell Labs
Institute of Scientific Information, Phila., PA
GM Research labs, Warren, MI
Honeywell

2. *Principal U.S. Government Agencies Funding NLP Research*
   ONR (Office of Naval Research)
   NSF (National Science Foundation)
   DARPA (Defense Advanced Research Projects Agency)

3. *Commercial NLP Systems*
   Artificial Intelligence Corp., Waltham, Mass.
   Cognitive Sytems Inc., New Haven, Conn.
   Symantec, Sunnyvale, CA
   Texas Instruments, Dallas, TX
   Weidner Communications, Inc., Provo, Utah
   Savvy Marketing International, Sunnyvale, CA
   ALPS, Provo, UT

4. *Non-U.S.*
   U. of Manchester, England
   Kyoto U., Japan
   Siemens, Corp., Germany
   U. of Strathclyde, Scotland
   Centre National de la Recherche Scientifique, Paris
   U. di Udine, Italy
   U. of Cambridge, England
   Phillips Res. Labs, The Netherlands

M. Forecast

Commercial natural language interfaces (NLI's) to computer programs and data base management systems are now becoming available. The imminent advent of NLI's for micro-computers is the precursor for eventually making it possible for virtually anyone to have direct access to powerful computational systems.

As the cost of computing has continued to fall, but the cost of programming hasn't, it has already become cheaper in some applications to create NLI systems (that utilize subsets of English) than to train people in formal programming languages.

Computational linguists and workers in related fields are devoting considerable attention to the problems of NLP systems that understand the goals and beliefs of the individual communicators. Though progress has been made, and feasibility has been demonstrated, more than a decade will be required before useful systems with these capabilities will become available.

One of the problems, in implementing new installations of NLP systems, is gathering information about the applicable vocabulary and the logical structure of the associated data bases. Work is now underway to develop tools to help automate this task. Such tools should be available within 5 years.

For text understanding, experimental programs have been developed that "skim" stylized text such as short disaster stories in newspapers (DeJong, 1982). Despite the practical problems of sufficient world knowledge and the extension of language required, practical tools emerging from these efforts should be available to provide assistance to humans doing text understanding within this decade.

The NRL Computational Linguistic Workshop (1981) concluded that text generation techniques are maturing rapidly and new application possibilities will appear within the next five years.

The NRL workshop also indicated that:

> Machine aids for human translators appear to have a brighter prospect for immediate application than fully automatic translation; however, the Canadian French-English weather bulletin project is a fully automatic system in which only 20% of the translated sentences require minor rewording before public release. An ambitious common market project involving machine translation among six European languages is scheduled to begin shortly. Sixty people will be involved in that undertaking which will be one of the largest projects undertaken in computational linguistics.* The panel was divided in its forecast on the five year perspective of machine translation but the majority were very optimistic.

Nippon Telegram and Telephone Corp. in Tokyo has a machine translation AI project underway. An experimental system for translating from Japanese to English and vice versa is now being demonstrated. In addition, the recently initiated Japanese Fifth Generation Computer effort has computer-based natural language understanding as one of its major goals.

In summary, natural language interfaces using a limited subset of English are now becoming available. Hundreds of specialized systems are already in operation. Major efforts in text understanding and machine translation are underway, and useful (though limited) systems will be available within the next five years. Systems that are heavily knowledge-based and handle more complete sets of English should be available within this decade. However, systems that can handle unrestricted natural discourse and understand the motivation of the communicators remain a distant goal, probably requiring more than a decade before useful systems appear.

REFERENCES

• Arden, B. W. (ed.) *What Can Be Automated*, Cambridge, MA: M.I.T. Press, 1980.

• Barr, A. and Feigenbaum, E. A., Chapter 4, "Understanding Natural Language," *"The Handbook of Artificial Intelligence,* Los Altos, CA: W. Kaufman, 1981, pp. 223-321.

• Burton, R. R., "Semantic Grammar: An Engineering Technique for Constructing Natural Language Understanding Systems," BBN Report 3453, BBN, Cambridge, Dec. 1976.

• Charniak, E. "Six Topics in Search of a Parser: An Overview of AI Language Research," IJCAI-81, pp. 1079-1087.

• Charniak, E. and Wilks, Y., *Computational Semantics,* Amsterdam: North Holland, 1976.

• Chomsky, N., *Syntactic Structures,* The Hague: Mouton, 1957.

• DeJong, G., "An Overview of the FRUMP System," *In Strategies for Natural Language Processing,* W. G. Lehnert and M. H. Ringle (eds.), Hillsdale, N.J.: Lawrence Erlbaum, 1982, pp. 149-176.

*EUROTA — A machine translation project sponsored by the European Common Market — 8 countries, over 15 universities, $24 M over several years.

• Dowty, R. et al., *Introduction to Montague Semantics,* Reidel, 1981.

• Fillmore, C., "Some Problems for Case Grammar," In R. J. O'Brien (Ed.), *Report of the Twenty-Second Annual Round Table Meeting on Linguistics and Languages Studies,"* Wash., D.C.: Georgetown U. Press, 1971, pp. 35-56.

• Finin, T. W., "The Semantic Interpretation of Compound Nominals," Ph.D. Thesis, U. of IL, Urbana, 1980.

• Gawron, J. M. et al., "Processing English with a Generalized Phrase Structure Grammar," *Proc. of the 20th Meeting of ACL,* U. of Toronto, Canada, 16-18 June 1982, pp. 74-81.

• Gazdar, G., "Unbounded Dependencies and Coordinate Structure," *Linguistic Inquiry,* 12, 1981, pp. 155-184.

• Gevarter, W. B., *An Overview of Computer-Based Natural Language Processing,* NBSIR 83-2687, National Bureau of Standards, Wash, D.C., April 1983 (also NASA TM 85635).

• Graham, N., *Artificial Intelligence,* Blue Ridge Summit, PA: TAB Books, 1979.

• Hendrix, G. G., Sacerdoti, E. D. Sagalowicz, D. and Slocum, J., "Developing a Natural Language Interface to Complex Data," *ACM Transactions on Database Systems,* Vol. 3, No. 2, June 1978.

• Hendrix, G. G. and Sacerdoti, E. D., "Natural-Language Processing: The Field in Perspective," *Byte,* Sept. 1981, pp. 304-352.

• Kaplan, S. J., (Ed.), "Special Section — Natural Language," *SIGART NEWSLETTER* #79, Jan 1982, pp. 27-109.

• McDonald D. D., "Natural Language Production as a Process of Decision-Making Under Constraints," Ph.D Thesis, M.I.T., Cambridge, 1980.

• Nishida, T. and Doshita, S., "An Application of Montague Grammar to English-Japanese Machine Translation," *Proc. of Conf. on Applied NLP,* Santa Monica, Feb. 1983.

• Reiger, C. and Small, S., "Word Expert Parsing," *Proc. of the Sixth International Joint Conference on Artificial Intelligence,* 1979, pp. 723-728.

• Schank, R. C. and Abelson, R. P., *Scripts, Plans, Goals and Understanding,* Hillsdale, N.J.: Lawrence Erlbaum, 1977.

• Schank, R. C. and Riesbeck, C. K., *Inside Computer Understanding,* Hillsdale, N.J.: Lawrence Erlbaum, 1981.

• Slocum, J., "A Practical Comparison of Parsing Strategies for Machine Translation and Other Natural Language Purposes," Ph.D Thesis, U. of Texas, Austin, 1981.

• Tennant, H., *Natural Language Processing,* New York: Petrocelli, 1981.

• Winograd, T., *Language as a Cognitive Process,* Vol I; Syntax, Reading, Mass: Addison-Wesley, 1983.

• "Applied Computational Linguistics in Perspective," NRL Workshop at Stanford University, 26-27 June 1981. (Proceedings in *Amer. J. of Computational Linguistics,* Vol. 8, No. 2, April-June 1982, pp. 55-83).

• Association for Computational Linguistics (ACL) Annual Conference.

• Computational Linguistics Conferences. (Biannual).

IV. SPEECH RECOGNITION AND SPEECH UNDERSTANDING

A. Introduction

Speech is our fastest means of discourse communication, being about twice as fast as the average typist. It is also nearly effortless: speech doesn't need visual or physical contact and it places few restrictions on the use of the hands or the mobility of the body. Speech is thus well suited to communication with a machine when the individual is engaged in other activities. Its effortlessness also makes it desirable for operating a computer, and it is a long term candidate for direct text preparation (automatic dictation).

Speech understanding systems have all the difficulties of natural language understanding plus the problem of interpreting the speech signal with all its noise and variability. As a result, speech understanding is one of the most difficult AI subjects, being a perception task related to the scene understanding problem in computer vision. Though the constraining aspects of natural language help reduce the magnitude of the task, it remains a major problem area.

Speech systems can be categorized into speech recognition systems and speech understanding systems, the former task being considerably easier. In addition, the systems further divide into those that work with isolated words and those that can handle connected speech, the latter being perhaps an order of magnitude more difficult than the former.

Finally, speech systems are also classified as speaker dependent and speaker independent. The former systems must be trained to recognize the particular speakers using it.

The heart of the speech problem (that gives rise to the above classifications) is the difficulty of recognizing the speech signal, but before we explore that area, let us briefly look at applications for speech devices.

B. Applications

There are many applications emerging for speech recognition and speech understanding systems. Some of these are listed in Tables IV-1 and IV-2.

C. The Nature of Speech Sounds:

It is beginning to be realized that acoustics and phonetics may be the key to speech understanding. Zue (1981) argues that human spectrograph-reading experiments indicate that phonetic recognition in speech systems can be improved substantially, which would result in much more capable speech systems.

Speech recognition is based primarily on the identification of words. An adult speaker may know 100,000 of the 300,000 words in the English language. Each language has a basic set of speech sounds called phonemes. In English there are only about 40 phonemes, compared with some 10,000 for the next largest speech unit, the syllable.

The sounds that make up human speech are generated by the flow of air through the vocal tract in three ways (Levinson and Liberman, 1981):

TABLE IV-1. Speech Recognition Applications.

Manufacturing Processes and Control

- Quality control data entry into computers
- Shipping and receiving — record entry, package sorting
- Maintenance and repair orders — part availability, work needed or under way.
- CAD/CAM

Office Automation

- Executive work station
- Word processing
- Data entry
- Control functions

Technical Data Gathering

- Cartography — inputs when working with maps.
- Working with blueprints
- Medical applications:
  - Dental records
  - Pathology
  - Services for the handicapped
  - Operating room logging
  - Command/control of medical instrumentation

Security Applications

- Building access
- Computer file access
- Communications security
- Speaker verification/identification

Consumer Products Applications

- Control functions
- Status queries

Equipment Subsystem Operation

- Aircraft
- Spacecraft
- Military equipment

TABLE IV-2. Speech Understanding Applications

- Universal access to large data bases via the telephone network.
- Automatic telephone transaction systems — Airline reservations and inquiries.
- Command and Control
  - — Military
  - — Business
- Operation of complex machines.

1. The vocal cords can be made to vibrate, resulting in the frequency of the sound referred to as pitch.
2. A constriction can be formed in the vocal tract, narrow enough to cause turbulence, resulting in noise-like sounds, like that used to produce "f".
3. Pressure built up behind a closure (such as the lips) can release a burst of acoustic energy as in the pronunciation of consonants, such as "p", "t" and "k".

These three sources of speech sound are shaped acoustically by the time-varying physical shape of the vocal tract.

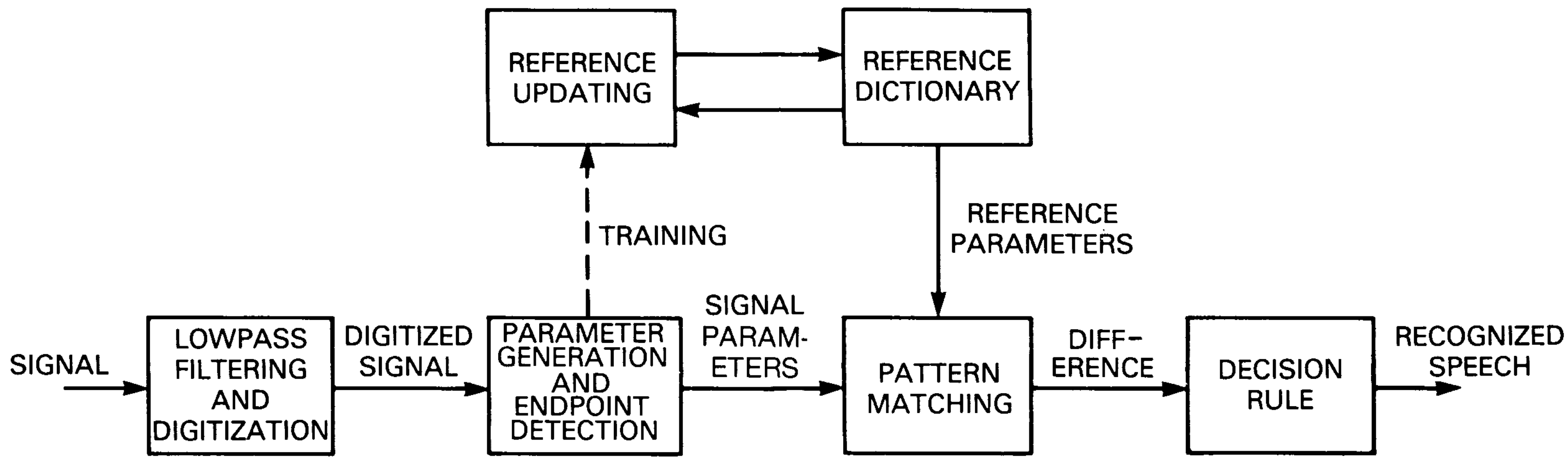
One way to characterize the speech signal is by its Fourier transform, which specifies the amplitude and phase of each of the frequencies in the Fourier series of the signal. As the phase makes little perceptual difference, the signal is represented in practice by its amplitude spectrum, in a representation called a spectograph.

D. Isolated Word Recognition

Figure IV-1 indicates a basic paradigm for speech recognition. The signal is first operated upon to emphasize the 2 to 3 kHz frequency range, filtered to chop off high frequencies (>8 kHz), then digitized. The end points of the word are detected, and a set of parameters representing the word are generated. This is then matched with stored parameter sets in the system's vocabulary, and the word with the closest match chosen. For a word, the acoustic signal varies both in duration and amplitude each time the same speaker says it. Thus it may have to be warped to achieve the best comparison with the reference — this task being one of the toughest problems for a speech recognizer. The warping is usually accomplished by dynamic programming.

Doddington and Schalk (1981, p. 28) state that:

> The most common means of feature extraction is direct measurement of spectrum amplitude, with, for example, a set of 16 bandpass filters. Another means is measurement of the zero-crossing rate of the signal in several broad frequency bands to give an estimate of the formant [resonant] frequencies in these bands. Yet another means is representing the speech signal in terms of the parameters of a filter whose spectrum best fits that of the input speech signal. This technique known as linear predictive coding (LPC) has gained popularity because it is efficient, accurate, and simple.



After Zue (1982, p51)

Figure IV-1. Basic Speech-Recognition Paradigm

E. Recognizing Continuous Speech

For continuous speech, rather than attempting to match all possible word patterns, it is often more efficient to work with speech units much smaller than words, particularly phonemes. Breaking down the speech signal into these smaller components and giving them symbols, is referred to as segmentation and labeling. Usually, several phoneme labels are assigned to each segment by a pattern-matching process, which also assigns a probability value representing the goodness of the match. With the appropriate acoustic-phonetic knowledge, it is possible to combine, regroup, and delete segments to form larger phoneme units. The lexical knowledge of word pronunciation can now be used to generate a multiplicity of word hypotheses. For a sufficiently limited vocabulary, and perhaps also employing some syntactic and word boundary knowledge, speech recognition can be achieved.

F. Speech Understanding

Arden (1980, pp. 475, 478) observes that:

> Speech-understanding systems differ somewhat from recognition systems, in that they have access to and make effective use of task-specific knowledge in the analysis and interpretation of speech. Further, the criteria for performance are somewhat relaxed, in that the errors that count are not the errors in speech recognition, but errors in task accomplishment.
>
> To successfully decode the unknown utterance, a speech perception system must effectively use the many diverse sources of knowledge about the language, the environment, and the context. These sources of knowledge include the characteristics of speech sounds (acoustic-phonetic), variability in pronunciation (phonology), the stress and intonation patterns of speech (prosodics), the sound patterns of words and sentences (lexicon), the grammatical structure of language (syntax), the meaning of words and sentences (semantics), and the context of the conversation (pragmatics) . . .
>
> What makes speech perception a challenging and difficult area of A.I. is the fact that error and ambiguity permeate all the levels of the speech-decoding process . . . .
>
> The grammatical structure of sentences can be viewed principally as a mechanism for reducing search by restricting the number of acceptable alternatives . . . .

Barr and Feigenbaum (1981, p. 332) note that the types of knowledge at the various levels in processing spoken knowledge include (from the signal level up):

1. *Phonetics* — representations of the physical characteristics of the sounds in all of the words in the vocabulary.
2. *Phonemics* — rules describing variations in pronunciation that appear when words are spoken together in sentences (coarticulation across word boundaries, "swallowing" of syllables, etc.);
3. *Morphemics* — rules describing how morphemes (units of meaning) are combined to form words (formation of plurals, conjugations of verbs, etc.);
4. *Prosodics* — rules describing fluctuation in stress and intonation across a sentence;
5. *Syntax* — the grammar or rules of sentence formation resulting in important constraints on the number of sentences (not all combinations of words in the vocabulary are legal sentences);
6. *Semantics* — the "meaning" of words and sentences, which can also be viewed as a constraint on the speech understander (not all grammatically legal sentences have a meaning — e.g., The snow was loud); and, finally,
7. *Pragmatics* — rules of conversation (in a dialogue, a speaker's response must not only be a meaningful sentence but also be a reasonable reply to what was said to him). For instance, it is pragmatic knowledge that tells us that the question "Can you tell me what time it is?" requires more than just a Yes or No response.

Using this knowledge, the hierarchical structure leading to speech understanding can be characterized as shown in Figure IV-2.

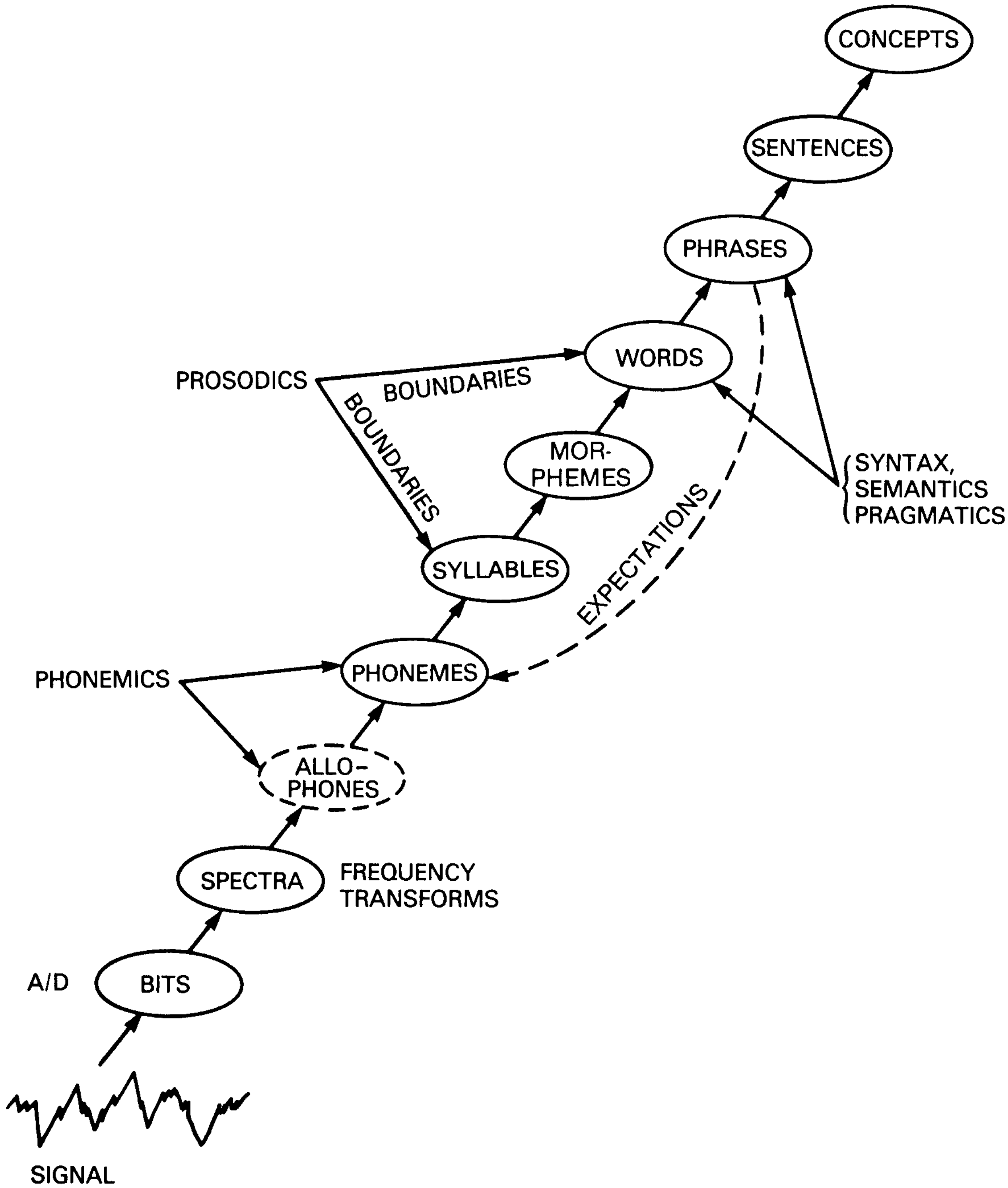


Figure IV-2. The Processing Hierarchy in Speech Understanding

G. The ARPA Speech Understanding Research (SUR) Project

1. Introduction

In 1971, ARPA (The Advanced Research Projects Agency) initiated a five year speech understanding research effort that proved to be one of the most significant projects in AI history. Not only did it greatly advance our knowledge of speech, but it also provided new insights on how to structure and control a complex "expert system."

Lea and Shoup (1979) reported that the ARPA SUR project had the highly ambitious goals of understanding, with 90% accuracy, continuous speech from a 1000 word vocabulary spoken by several cooperative speakers under near ideal conditions of quiet rooms and high-fidelity equipment. It was intended that the processing take no more than several times real-time using large very fast computers.

There were three principal complete systems developed under the project — HEARSAY II and HARPY at Carnegie Mellon University (CMU), and HWIM (Hear What I Mean) at Bolt, Berenek and Newman (BBN). In 1976, the ARPA goals were essentially met at CMU by HARPY exhibiting a 95% accuracy and HEARSAY II achieving a 90% accuracy. HWIM had a substantially lower accuracy, but utilized a more difficult vocabulary. (HWIM's domain was Travel Budget Management. HEARSAY II's and HARPY's was Retrieval of AI Documents.) These three systems were heavily knowledge-based and are now considered to be expert systems.

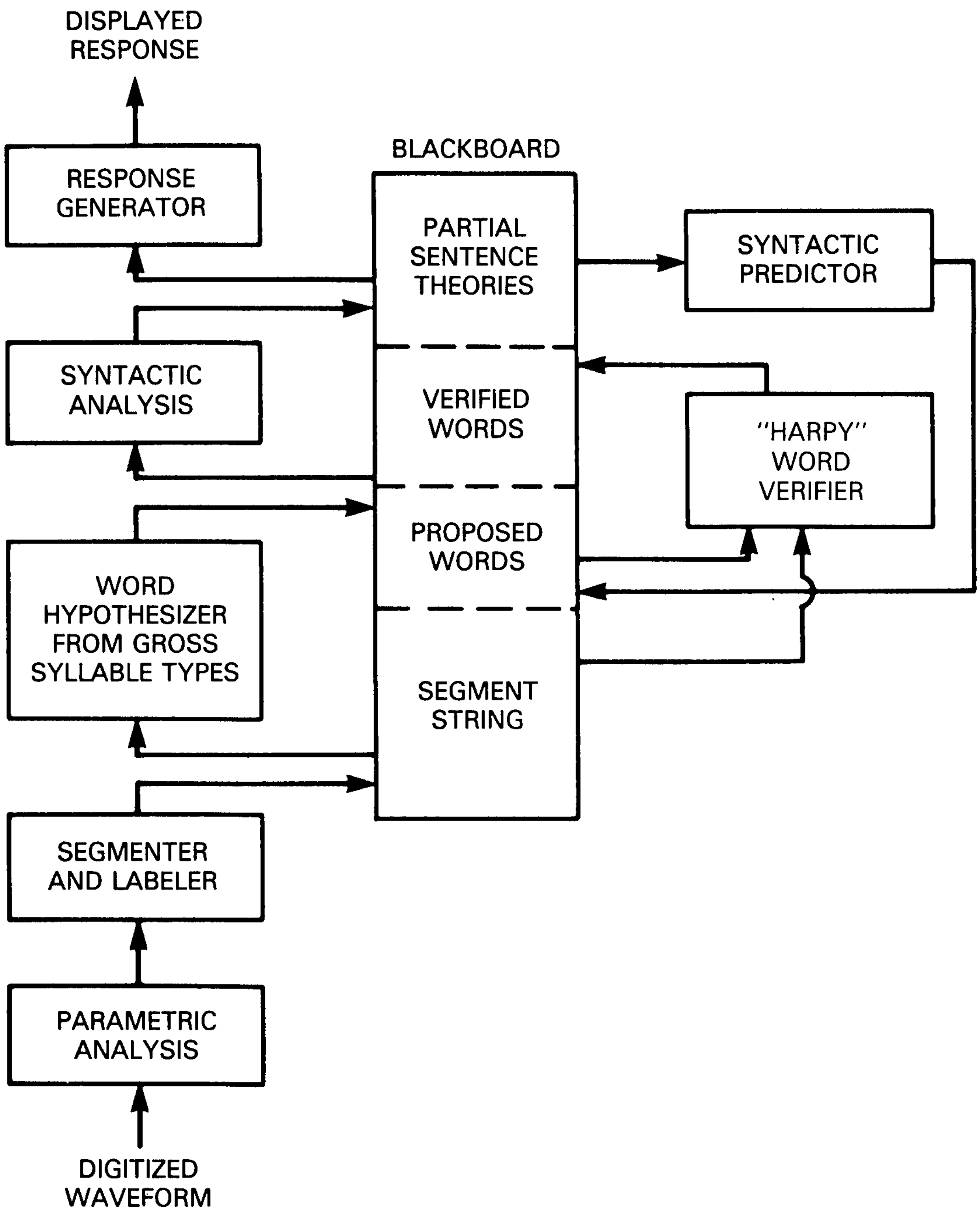
All the ARPA SUR systems utilized a combination of bottom-up and top-down processing. The lower levels used knowledge about the variable phonetic composition of the words in the vocabulary (lexicon) to interpret pieces of the speech signal by comparing it with prestored patterns. The top level aided in recognition by building expectations about which words the speaker was likely to say, using syntactic and semantic constraints (Barr and Feigenbaum, 1982, pp. 326-327).

2. HEARSAY II

HEARSAY II is characterized by its cooperative problem-solving system architecture (see Figure IV-3) which employs a set of programmed "specialists" (Knowledge Sources: KS's) interacting via a shared common blackboard on which their decisions were recorded. The blackboard can be visualized as a global data structure representing a multi-level network of alternative hypotheses.

HEARSAY has a total of 12 KS's, which at the lower levels created syllable class hypotheses from segments, word hypotheses from syllables, etc. At the higher levels, KS's acted to: predict all possible words that might syntactically precede or follow a phrase, create phrase hypotheses from verified contiguous word-phrase pairs, etc.

The majority of the hypotheses contributed by the KS's at any level did not end up in the final interpretation of the sentence. Instead, only the most likely hypotheses were chosen for expansion. The individual KS's operated somewhat independently and asynchronously through pattern-invoked programs when matching patterns appeared on the blackboard. To economize on computing resources, each hypothesis was rated and (using an appropriate scheduling routine) the most likely patterns were expanded first.



After Klatt (1977)

Figure IV-3. A Block Diagram of the CMU Hearsay-II System Organization

3. *HARPY*

A crude way of thinking of HARPY is as a compiled version of HEARSAY II. HARPY uses a single precompiled network knowledge structure. Barr and Feigenbaum (1981, p. 349) report:

> The network contains knowledge at all levels: acoustic, phonemic, lexical, syntactic, and semantic. It stores acoustic representations of every possible pronunciation of the words in all of the sentences that HARPY recognizes. The alternative sentences are represented as paths through the network, and each node in the network is a template of allophones (distinctive variations of phonemes, dependent on adjacent phonemes).
>
> The paths through the network can be thought of as "sentence templates," much like the word templates used in isolated-word recognition.

HARPY uses a heuristic method called "beam search" for searching for the sentence in the network that most closely matches the input signal. HARPY proceeds from left to right through the network, matching spoken sounds to allophonic states; and assigning scores based on the goodness of the match. HARPY keeps the paths with the best cumulative scores, pruning away others which fall some threshold amount below the best scoring path (Erman et al, 1980).

4. *HWIM*

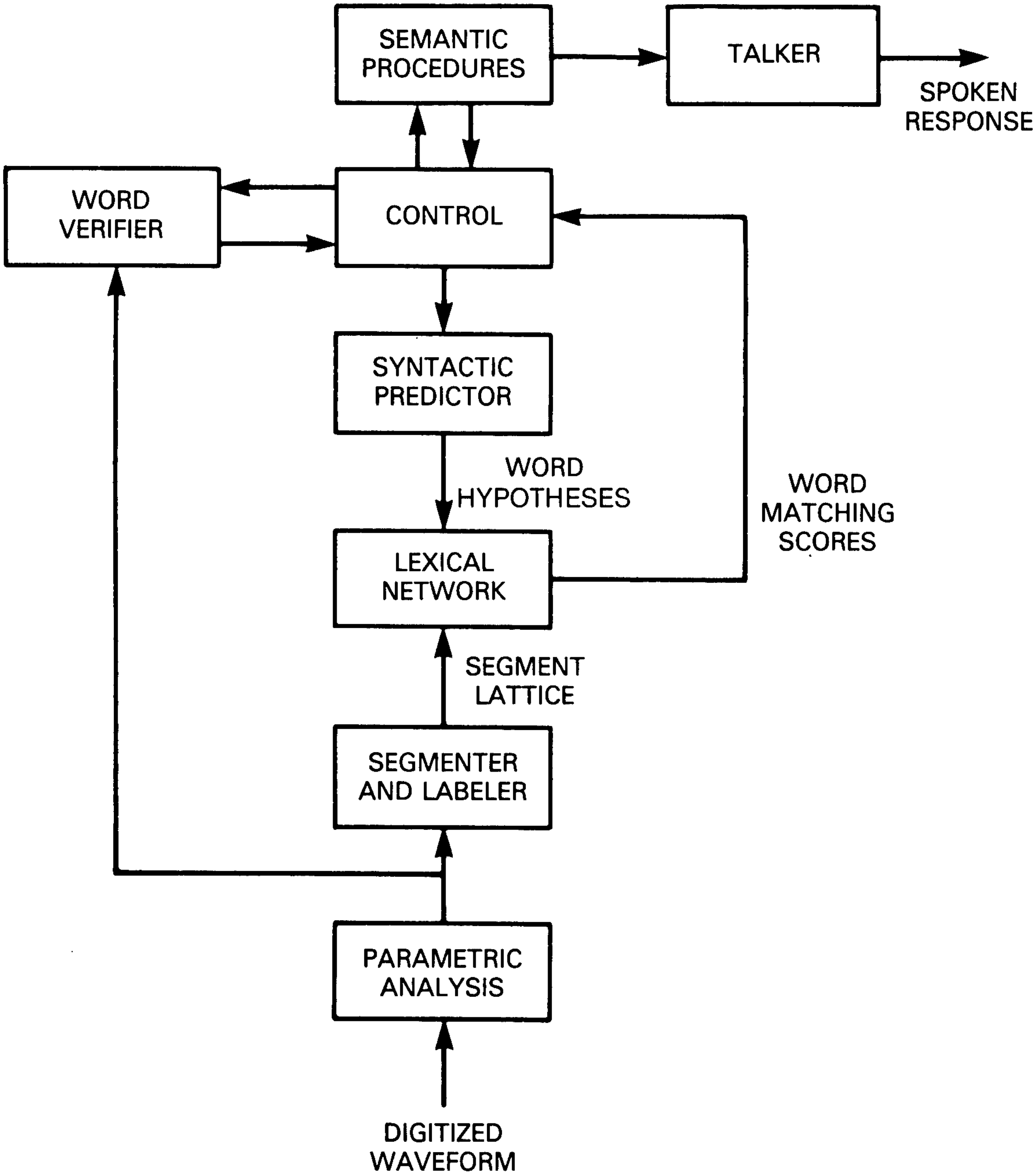
The HWIM (Hear What I Mean) speech understanding system was developed at BBN. HWIM's domain was that of travel budget management. HWIM's organization is shown in Figure IV-4. The lower components digitize the speech signal and generate a parametric representation of it, which is then segmented and labeled into phonemes which are ranked as to the quality of their match. These ranked phonemes are pictured as a segmented lattice, which is a graph that is divided into time segments and read from left to right. This graph is matched against a dictionary of work pronunciations (stored as a network with phonemes for nodes) by lexical retrieval components which generate word hypotheses.

HWIM's higher levels include information about trips (semantics), syntax and word verification. The verification component takes the pronunciation of hypothesized words and generates a synthesized parameter representation that is compared to the parameters generated from the actual signal.

HWIM has a central control which uses the system's knowledge sources as subroutines. The system extends bottom-up theories using the top-down syntactic and semantic components. The system expands its hypotheses about the first recognized word in the sentence.

5. *Summary of the ARPA SUR Program*

ARPA's program did not result in a useable speech understanding system. The resulting systems were too slow, too restricted and required large computational resources. However, it did discover and elucidate much new information about speech, and developed new architectural insights, particularly the blackboard architecture that has since been used in other AI systems. Performances of the different systems were difficult to compare because of the different vocabularies and domains employed. One critical factor in comparison is the average branching factor (ABF). This refers to the average number of words that might come next after each work in a legal sentence. Table IV-3 summarizes the three major ARPA SUR projects. Note that the ABF is 196 for HWIM's database retrieval task, versus 33 for HEARSAY's and HARPY's document retrieval task.



After Klatt (1977)

Figure IV-4. Block Diagram of the BBN HWIM System Organization

TABLE IV-3. Summary of ARPA's Speech Understanding Systems

Name/Org	Domain/ Purpose	Approach	Knowledge Rep.	Control	ABF	Accuracy	Comments
HEARSAY II CMU	AI Publications Document Retrieval	Utilizes cooperating independent system experts (Knowledge Sources) that communicate via posting hypotheses on a blackboard.	Independent KS's composed of production rules.	• Asynchronous pattern-invoked knowledge sources • Opportunistic scheduling by first expanding the highest scoring hypothesis	33	90%	• Development of blackboard architecture and use of independent cooperating knowledge sources most significant • A parallel processor version has been built to exploit KS modularity.
HARPY CMU	AI Publications Document Retrieval	Compiled a network of all possible pronunciations of all possible sentences. Paths thru network are "sentence templates."	Precompiled Network. Each node is a template of allophones, which when linked form acoustic representations of every possible pronunciation of words in the domain.	"Beam Search" No backtracking	33	95%	• Approach cannot easily accommodate pragmatics. • Needs a large memory. • Sensitive to missing acoustical segments and missing words.

TABLE IV-3. Summary of ARPA's Speech Understanding Systems (cont.)

Name/Org	Domain/ Purpose	Approach	Knowledge Rep.	Control	ABF	Accuracy	Comments
HWIM BBN	Travel Budget Management	• The system extends bottom-up word theories, using top-down syntactic and semantic components. Verifies hypothesized words by generating a parameter representation that is compared with that from the actual speech input. • Uses an ATN semantic grammar.	Uses networks to represent: 1) trip facts and relations. 2) Lexicon 3) Phoneme hypotheses from signal	• Centralized control using KS's as subroutines. • Expands sentences about the first recognized word in sentence (Island Driving).	196	44%	• Speaker Independent • Very slow • Most difficult domain in SUR project.

H. State of the Art

1. Speech Recognition

Table IV-4 is a summary of a recent Texas Instruments' study of commercial speech recognizers tested on a 20 word vocabulary consisting of the 10 spoken digits "zero" thru "nine" and ten command words: start, stop, yes, no, go, help, erase, rubout, repeat and enter.

In 1982, speaker-dependent connected-word short-string, small vocabulary (approx. 50 words) recognizers were commercially available. These could recognize up to 90 wpm of connected speech compared to a typical person's speaking rate of 150 wpm. The vocabulary size is usually less than 150 words, but is application dependent. Recognition accuracies of 98% or greater are being achieved in factory environments. Current turnkey systems are in the $5K to $75K range. Consumer product speech-recognizer subsystems for toys, personal computers, voice-controlled appliances, etc., cost from $6 to $100.

Voice recognition systems are here, viable, proven, but still somewhat costly. In industry applications, they have demonstrated large increases in productivity. Hundreds of successful installations exist today. Plohar (1983) discusses the human factor considerations associated with successful applications.

2. Speech Understanding

There are no commercial true speech-understanding systems today. However, there are a number of U.S. companies working on future commercial systems.

a. Bell Labs

Has been working on a semantic sentence recognizer and interpreter utilizing a finite state grammar and a small vocabulary. The intent is to produce an interactive speech understanding system for use over the telephone (Levinson and Liberman, 1981).

b. IBM — T.J. Watson Res. Center

IBM has had the largest effort in continuous-speech recognition and understanding, capitalizing on the HARPY "Beam Search" approach.

c. Other organizations involved in developing speech understanding systems include BBN.

I. Who Is Doing Speech Recognition Related Work

1. Commercial Organizations

IBM
TI
Bell Labs
Verbex
Nippon Electric
Threshold Technology
Interstate Electronics
Matsushita
Scott Instruments
Sanyo
INTEL
ITT (San Diego)
Fairchild

TABLE IV-4. T.I.'s Test of Speech Recognizers on Individual Words

Manufacturer	Model*	Nominal Price in 1981	Nominal Price for Comparable 1983 Model	% Substitutions
Verbex	1800	$65K	$19.6K	0.2
Nippon Electric	DP-100	$65K	$27K	1.2
Threshold Technology	T-500	$12K	$5K	1.4
Interstate Electronics	VRM	$2.4K	$2.4K	2.9
Heuristics	7000	$3.3K	NA	5.9
Centigram	MIKE 4725	$3.5K	NA	7.1
Scott Instruments	VET/1 (home computer peripheral)	$ .5K	$ .9K	12.6

*First two systems are capable of connected speech.
Verbex is the only system having speaker-independent capability.

After Doddington and Schalk (1981).

Hewlett Packard
Haskins Lab
Lincoln Labs
Speech Communications Research Lab
Sperry Univac
Votan
Voice Machine Communications
Voice Processing Corp.
General Instrument — Milton Bradley
Voice Control Systems

2. *Universities*
M.I.T.
C.M.U.
V.P.I.
U of CA at Berkeley

J. Problems and Issues*

- Speech perception at the acoustic level is a critical factor in achieving advanced recognition capability. Current commercial word recognizers have not yet made full use of available knowledge.
- Widespread use of speech recognizers await the availability of low cost connected-speech systems achieving better than a 99% accuracy with limited vocabularies — 100 words.
- Capabilities of a word recognizer depend on:
  (1) Can it recognize connected speech?
  (2) Is it speaker independent?
  (3) How big a vocabulary can it recognize?
- The greatest difficulty that speech recognizers have is determining word end-points — the source of many word-recognition errors for isolated word recognizers.
- A major problem is separating linguistically significant variations in the speech signal from insignificant variations (such as variations in word pronunciations).
- Noise is also a major problem in speech recognition, often resulting from actions of the speaker himself.
- Large vocabulary size is a problem to users, who need to remember what the machine can recognize.
- The two main errors made by speech recognizers are:
  (1) Substitution, and
  (2) Rejection
- Other less common errors are insertion and deletion.
- There are as yet no standards for test or evaluation of systems — a major problem.

*These have been gleaned primarily from Doddington and Schalk (1981).

- It is not the number of words that are the major difficulty, it is how close their sound is to each other. The natural English alphabet is a particularly difficult set of sounds to distinguish.
- Software or hardware may also have idiosyncracies that adversely affect recognition performance.
- As recognizer performance improves, evaluation becomes more difficult, because more testing is required to achieve statistical significance.
- The pronunciation of individual words change depending on the adjacent words in the sentence.
- The hypothesize and test approach needs abundant computer power — a major factor limiting its commercial use.
- Integrating recognizers into an application requires substantial software and human factors considerations. This has limited real-world adoption.

K. Future Trends

It is anticipated that speaker-independent, continuous-speech recognition systems with limited vocabularies (10-20 words), having an accuracy of 98% or better, will be available by the mid-1980's. Automatic dictation will probably not appear before the 1990's, with Japanese language systems being the first to appear. (Japanese language has only on the order of 500 syllables, compared to 10K for English.) Speech understanding is a major part of the Japanese 5th Generation Computer Project (Feigenbaum and McCorduck, 1983).

Due to the advancement in VLSI, it is expected that voice recognition chips for toys will soon be in the $6 range — $50 for a complete system.

A strong expectation is that a speech understanding system using a natural language parser will be introduced by IBM in the mid-80's.

Around 1990, true commercial speech understanding systems, having the capabilities of the ARPA SUR systems but operating in near real-time, are expected to appear.

By 1990, speech recognition and understanding is expected to be a billion dollar a year industry (Elphick, 1982).

REFERENCES

- Arden, B. W., (Ed.), *What Can Be Automated,* Cambridge: M.I.T. Press, 1980.
- Barr, A. and Feigenbaum, E. A. (Eds.), *The Handbook of Artificial Intelligence,* Vols I and II, Los Altos, CA: W. Kaufmann, 1981, 1982.
- Doddington, G. R. and Schalk, T.B., "Speech Recognition: Turning Theory to Practice," *IEEE Spectrum,* Sept. 1981, pp. 26-32.
- Elphick, M., "Unraveling the Mysteries of Speech Recognition," *High Technology,* Vol 2, No. 2, March/April 1982, pp. 71-76.
- Erman, L. D., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R., "The HEARSAY-II Speech-Understanding System: Integrating Knowledge To Resolve Uncertainty." *Computing Surveys,* Vol 12, No. 2, 1980.

- Feigenbaum E. A. and McCorduck, P., *The Fifth Generation,* Reading, Mass: Addison-Wesley, 1983.
- Klatt, D. H., "Review of the ARPA Speech Understanding Project," J. *Acoustical Soc. of America,* Vol 62, No. 6, Dec. 1977, pp. 1345-1366.
- Lea, W. A. and Shoup, J. E., *Review of the ARPA SUR Project and Survey of Current Technology in Speech Understanding,* Speech Communications Research Lab., Los Angeles, Jan. 16, 1979.
- Levinson, S. E. and Liberman, "Speech Recognition by Computer," *Scientific American,* Vol 244, No. 4, April 1981, pp. 64-76.
- Pluhar, K., "Speech Recognition — An Exploding Future for the Man-Machine Interface," *Control Engineering,* Jan. 1983, pp. 70-73.
- Zue, V. W., "Acoustic-Phonetic Knowledge Representation: Implications from Spectrogram Reading Experiments," *Proceedings of the 1981 NATO Advanced Summer Institute on Automatic Speech Analysis and Recognition,* D. Reidel Pub., 1981.

V. SPEECH SYNTHESIS

A. Introduction

Speech synthesis — speech output from a computer — is an emerging technology whose products are already becoming commonplace. Though the present market for these devices is still small, the future looks very bright.

Speech synthesis is not normally considered an AI topic, though it is sure to play an important part in many future AI systems, particularly when coupled with speech understanding. One may very well consider these synthesis systems, which employ rules (often heuristic) for deriving speech from stored speech elements, as an example of an "expert system on a chip."

B. Why Synthesis

One approach to making available speech when needed is to record the speech and play it back as required. The disadvantage is that mechanical devices are often unreliable and the ability to generate new sentences from stored words is quite limited because of access time, and therefore unsuitable for most computer-based applications.

A more reliable approach is to use digital sound recording techniques, enabling speech to be stored in solid-state memories having no moving parts to break down. The disadvantage is that an enormous amount of storage is required — in the order of 50,000 bits per second of digital speech (at the typical speaking rate of 150 words per minute). However, if words are represented by the digital code for their letters, the same information requires only about 100 bits per second of speech. This two to three orders of magnitude difference highlights the importance of speech compression for any digital representation of speech, not only to save storage requirements, but also to vastly reduce the bandwidth required for electronic speech transmission. All speech synthesis methods use some form of speech compression.

Speech synthesis serves three basic purposes:

1) Recreating speech from a compressed speech representation
2) Generating speech from stored speech elements such as by concatenating representations for words, and
3) Generating speech from text.

The first purpose is associated with minimizing storage or transmission bandwidth requirements. The second with creating speech from stored components under microprocessor or computer control. The third with reading machines and computer-human interaction.

An indication of applications of speech synthesis is given in Table V-1.

C. Human Speech

As many speech synthesizers actually employ an approximate simulation of the human speech production mechanism, it is helpful to briefly review human speech and its generation. Human

TABLE V-1. Applications of Speech Synthesis.

Military

- Operation of military equipment
- Warnings
- Reminders
- Service and operation aids
- Trainers and simulators
- Secure communications

Computer

- Communication by computers to users.

Consumer

- Talking appliances
- Teaching devices
- Toys
- Talking typewriters and calculators
- Talking watches
- Automobile warning devices, reminders, and annunciators for instruments
- Devices for the blind
- Communication for the speech handicapped

Telecommunications

- Synthesized telephone messages
- Speech compression for "store and forward," to reduce communication costs
- Vocal delivery of electronic mail

Industrial

- Speaking instruments
- Speaking cash registers
- Alarm systems
- Automated office equipment
- Industrial process control
- Station and floor announcers for trains, buses, elevators, etc.
- Systems operations where the operators have their visual attention elsewhere
- Emergency warning devices for airplanes, machines, etc.
- Control room annunciators for sensors
- Text readers
- Data entry (with vocal verification)

speech consists basically of a combination of vocal sounds such as vowels, fricative sounds—such as f, th or sh, and plosive or stop consonant sounds such as b and d.

The human vocal tract can be considered as an acoustic tube terminated at one end by the vocal cords and at the other end by the lips. This resonant tube has a side branch — the nasal resonator — separated by a flap called the velum.

Voiced sounds are produced by forcing air from the lungs past the tensed vocal cords which are thus forced to vibrate, emitting puffs of air into the vocal tract. (The puff frequency — about 100 hertz in males, 200 hertz in females — is a function of the vocal cord size and tenseness.) These puffs of air excite the vocal tract, stimulating their resonant (formant) frequencies. Most of the resulting sound energy is contained in these resonant responses, the frequency of which can be varied by changing the shape of the vocal tract by moving the lips, jaw or tongue.

Fricative sounds occur when a constriction in the vocal tract leads to turbulent air flows after the constriction.

Plosives are generated by briefly closing the vocal tract until pressure builds up and then releasing the pressure.

D. Electronic Simulation of the Speech Mechanism

The three basic human speech sounds can be electronically simulated as follows — as illustrated by the Computalker Consultants Model CT-1* synthesizer shown schematically in Figure V-1.

Voiced sounds can be simulated by passing energy from a variable periodic source — corresponding to the vocal cord puffs — through a series of variable filters (f_1, f_2, f_3) corresponding to the vocal tract resonances (formants). Plosive sounds are produced the same way, but require rapid changes in the amplitude parameters A_o and A_n. Fricative sounds are produced by passing white noise through a variable filter (f_f). Some sounds, such as v and z, are produced using both the periodic and noise mechanisms.

Using this approach, human speech can be simulated by controlling the frequency parameters (f_i) and the amplitude parameters (A_i) over time. Some variant of this basic method — referred to as parametric coding—is used in all speech synthesizers that simulate human speech production.

E. Synthesis in Speech Compression and Regeneration

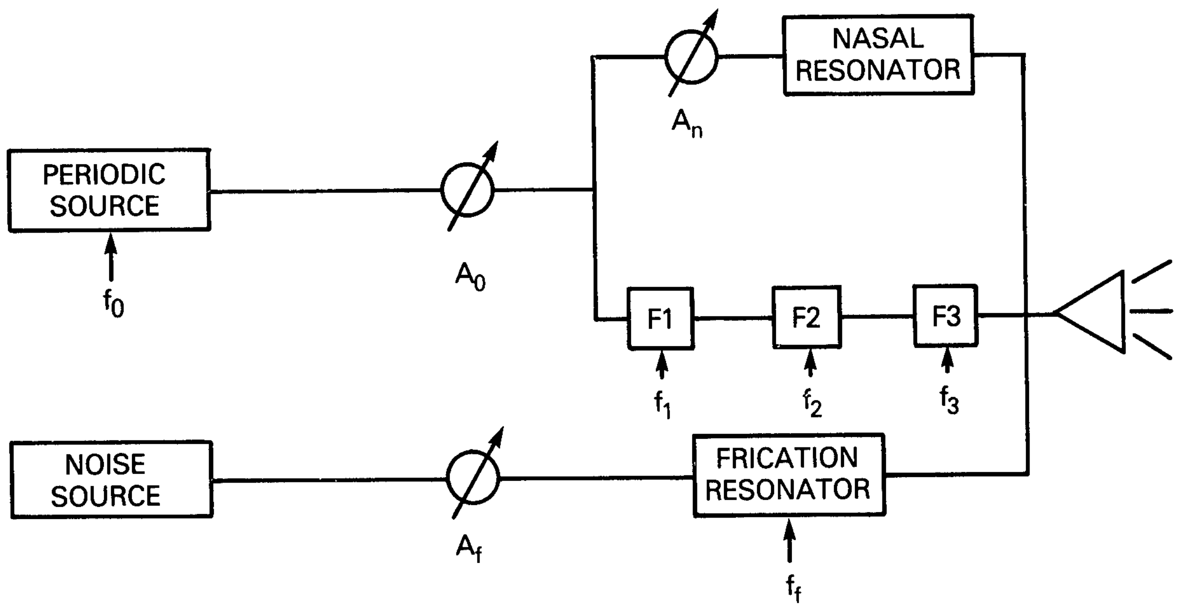
Synthesis has the role of regeneration in speech compression schemes (associated with speech storage or minimal bandwidth speech transmission).

There are two basic speech compression techniques — frequency domain analysis (parametric coding as discussed in the previous section on electronic simulation), and time domain analysis. Frequency domain methods tend to dominate commercial speech synthesis, but time domain analysis has become important for limited-vocabulary word synthesis.

The frequency domain approach analyzes the incoming speech to be compressed and generates the parameters needed for regenerating the signal using an electronic simulation of the vocal tract. In some cases, these parameters may be further compressed for reduced storage. Speech is generated by inverting the process as indicated in Figure V-2.

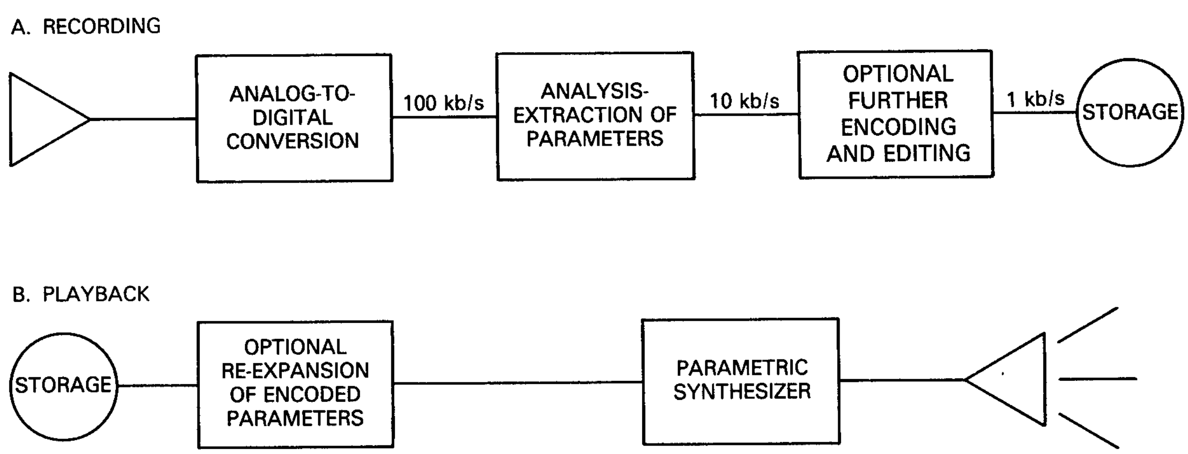
Time domain analysis is characterized by waveform compression techniques. Waveform digitization coding, researched extensively by Bell Labs, takes the original waveform of spoken

*No longer in production, but the Phillips speech chip essentially does the same thing.



After Sherwood (1979)

Figure V-1. A Simplified Diagram of the Computalker CT-1 Parametric Synthesizer.



After Sherwood (1979)

Figure V-2. Recording and Reproduction of Speech Using a Compressed-Speech System.

words and compresses them using a complicated algorithm. The final compressed waveform is stored as bits in memory for later reconstruction of the original waveform. Though generally producing better sounding speech than parametric coding, waveform digitization coding requires two to four times as much storage as that needed for parametric coding.

F. Parametric Coding Schemes

1. Introduction:

All frequency domain compression techniques employ some sort of electronic model of the human vocal tract. Thus, all have one or more filters to simulate vocal tract resonances, and periodic and noise energy sources, and are controlled by varying the parameters associated with pitch, loudness, and filter frequencies.

2. Formant Coding

This is a straightforward approach to controlling an electronic model of the vocal tract by controlling the tunable filters using parametric signals that represent the formant (vocal tube resonant) frequencies such as those shown in Figure V-1. As the formant frequencies change relatively slowly, the parameters need to be updated relatively infrequently, thus allowing data compression.

3. Linear Predictive Coding (LPC)

LPC, pioneered by TI for "Speak and Spell," is a form of formant coding which allows further compression of the parameters. As the formant frequencies tend to change slowly, current samples are predicted from weighted linear combinations of previous samples. TI's LPC's clever prediction approach, and the use of an ingenious lattice filter, greatly simplifies the synthesis circuitry. The resulting system can be stored on a single chip and produces high quality natural sounding speech.

4. PARCOR

PARCOR (partial correlation), utilized by Japanese manufacturers, is a variant of LPC. LPC extrapolates from a series of formant samples to predict following formant frequencies. Though most speech patterns change slowly, plosive and fricative sounds involve rapid changes. PARCOR makes LPC more sensitive to sudden changes by giving greater emphasis to the correlation between adjacent parametric samples and less to the longer term patterns. However, there appears to be little resultant subjective differences in observed speech quality between the two approaches.

5. Line Spectrum Pair (LSP)

NTT (Nippon Telephone and Telephone Public Corp.) which developed PARCOR, has come up with LSP, an approach allowing still further compression. LSP defines the boundary conditions for the individual formant frequencies as those corresponding to the open and closed vocal tract. NTT claims that for a complete system, some 40% more compression can be achieved with LSP than with PARCOR, while maintaining nearly the same speech quality.

6. Parametric Waveform Coding (PWC)

PWC is another variant of LPC, as used by Centigram's Voice Ware system to produce vocabularies for the Lisa Speech Board.* PWC uses a variable-length slice of waveform to produce the linear prediction coefficients. Each slice (about 20 milliseconds in length) corresponds to a "glottal event" — the event associated with each puff of air passing through the vocal tract. Voice Ware uses an array processor to determine 13 linear prediction coefficients for each glottal event. To synthesize speech, the Lisa Speech Board uses these coefficients and the lengths of the events to recreate speech waveforms as in other LPC synthesizers. The PWC approach tends to yield more natural speech than the simpler LPC systems, but requires a higher data rate.

G. Waveform Coding Schemes

1. ADPCM

Digitized speech at a 8 kHz sampling rate results in 32,000 bits per second (bps) for a 4 bit sampling size using the adaptive differential pulse code modulation (ADPCM) proposed as the worldwide preferred method of digitized voice telephone signals for long distance transmission. In ADPCM, the digitized speech is encoded in terms of the amplitude differences between adjacent samples. These differences are adaptively encoded in terms of quantization level (a function of the previous quantization level and the previous PCM value). A close relative of ADPCM is CVSD (continuous variable slope delta modulation).

2. Mozer's Waveform Coding

Though ADPCM is suitable for telephone transmission, its high bit rate is unsuitable for stored speech synthesis. A scheme by Dr. Forrest Mozer of the University of California is a variation of ADPCM which provides substantial further compression. This technique has been incorporated into the National Semiconductor's Corporation's Digitalker. Dr. Mozer's approach is to:

1) Analyze the waveform to detect short periods with little change. The waveform for these periods are then replaced with identical waveforms.
2) Fourier analyze the signal and adjust the phase angle of each Fourier component to produce a symmetrical waveform and then discard half.
3) Discard low amplitude portions of the waveform which are not heard by the ear.
4) Employ ADPCM to further reduce data.

The net result of these actions is more than a 40 to one reduction in the data that needs to be stared, as compared with the data in direct digitization. To produce speech the process is inverted. Though these resultant signals look little like the original, the result is very good speech reproduction.

H. Coding the Words To Be Stored

Though the schemes discussed thus far provide a huge amount of reduction in the storage required, generating the required custom vocabulary in terms of the stored parameters requires hand tailoring by an expert. As yet, there is no acceptable automatic mechanism for directly converting speech into satisfactory storage elements for encoding schemes that provide high data

*No longer in production.

compression. (ADPCM is automatic. Parametric schemes can be automated with small residual errors.)

Developing the vocabulary for the Mozer Waveform Coding, used in National Semiconductor's Digitalker, takes about one hour of processing per word. It involves working with the data compression and zero phase-encoding algorithms, that produce the stored bit patterns, making it very difficult for users to program their own custom vocabularies (Ciarcia, 1983).

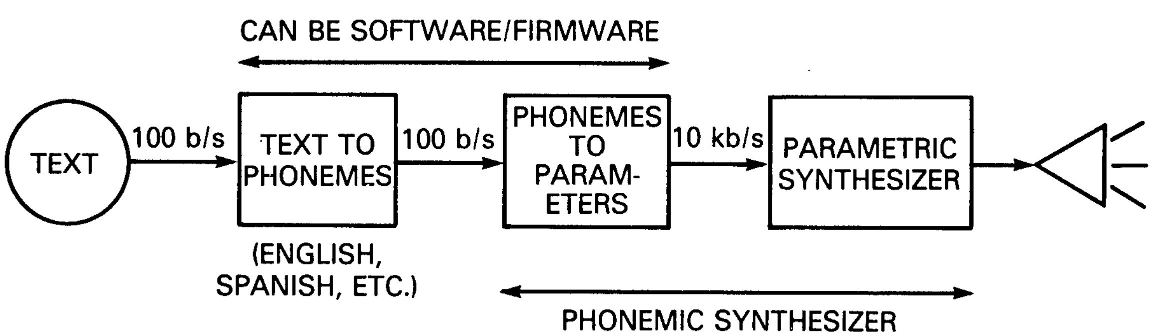
To enable users to develop their own custom vocabularies for their products, when large vocabularies are required, Centigram Corp. has offered as a product their Voice Ware development system. With it, users can input tape recorded voice to a digitizer that supplies a 4800 bps data stream to a microprocessor-based CRT-terminal work station. The station converts the signal into parametric waveform coding (PWC). The user can then edit the messages, combine them into files, and feed them back through the Lisa synthesizer to hear how they sound. If the sound is unsatisfactory, particularly for concatenated phrases, the phrases can be rerecorded to achieve the desired continuity and balance.

In general, for synthesizer users requiring a small custom vocabulary, it is customary for them to contract with the synthesizer manufacturer or other development source for the words required. This cost is in the order of $100 per word for LPC chips.

I. Generating Speech from Text

English has some 40 basic speech sounds called phonemes, corresponding to 16 vowel sounds, 6 stops, 8 fricatives, 3 nasals (such as ng), 4 liquids/glides (such as l in lice) and 3 others (such as ch in church). These sounds vary somewhat depending upon how they are combined into words or used in speech. These phoneme variations are called allophones. (Texas Instruments developed a set of 128 allophones to characterize English speech.) Allophones and the rules to string them together can be stored in computer memory chips. The first text-to-speech system used a phonemic synthesizer (Votrax). Votrax utilized a hard-wired phonemic to parameter converter which then fed a formant synthesizer to create speech. A simplified text-to-speech system schematic is given in Figure V-3.

The highly-intelligible state-of-the-art speech synthesizer, the Speech Plus "Prose 2000," utilizes a generation approach consisting of five serial processes: 1) Text normalization, 2)



After Sherwood (1979)

Figure V-3. Text to Speech Synthesis.

Phonemics, 3) Allophonics, 4) Prosodics, 5) Parameter generation. For words not in the exceptions lexicon, the phonemics process is implemented as a real-time expert system consisting of a small rule interpreter and an ordered set of about 400 context-sensitive rules.

J. State of the Art

Elphick (1981, p. 42) notes that:

> Most commercial synthesizers, especially low-cost ones used for consumer products, derive their speech elements from recordings of actual human speech. The recorded speech patterns are compressed, and the speech is disassembled into a vocabulary of small elements for later reassembly into messages.

High quality speech by phoneme synthesizers has been achieved in research systems, but not in commercial systems. The most natural commercial speech synthesizers use the waveform approach.

Figure V-4 is an indication of speech quality versus bit storage requirements for the various synthesis techniques.

Thus far, in industrial applications, only short messages are practical, as prolonged listening to synthetic speech tends to fatigue the operators (Andreiev, 1981).

Speech chips with limited vocabularies are available in the range of $10 and up. To construct the initial representations for new words (to be stored in ROMs) runs upward of tens of dollars per word.

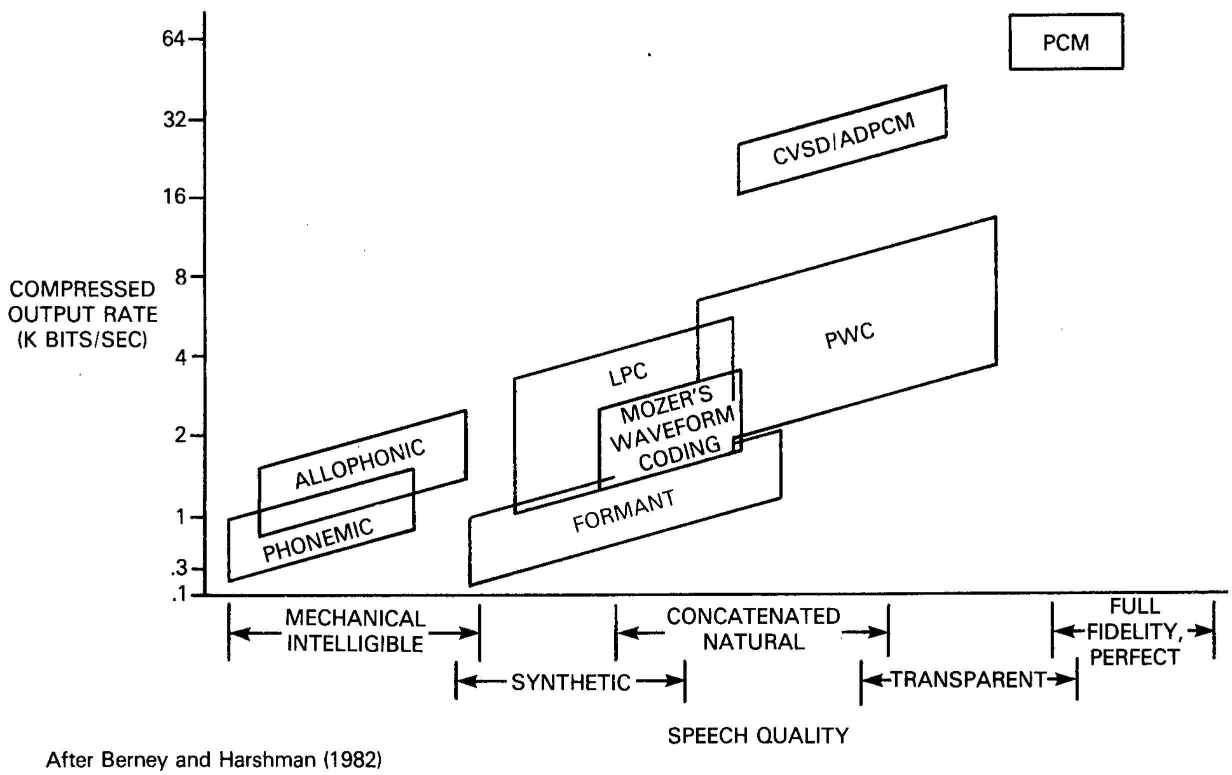
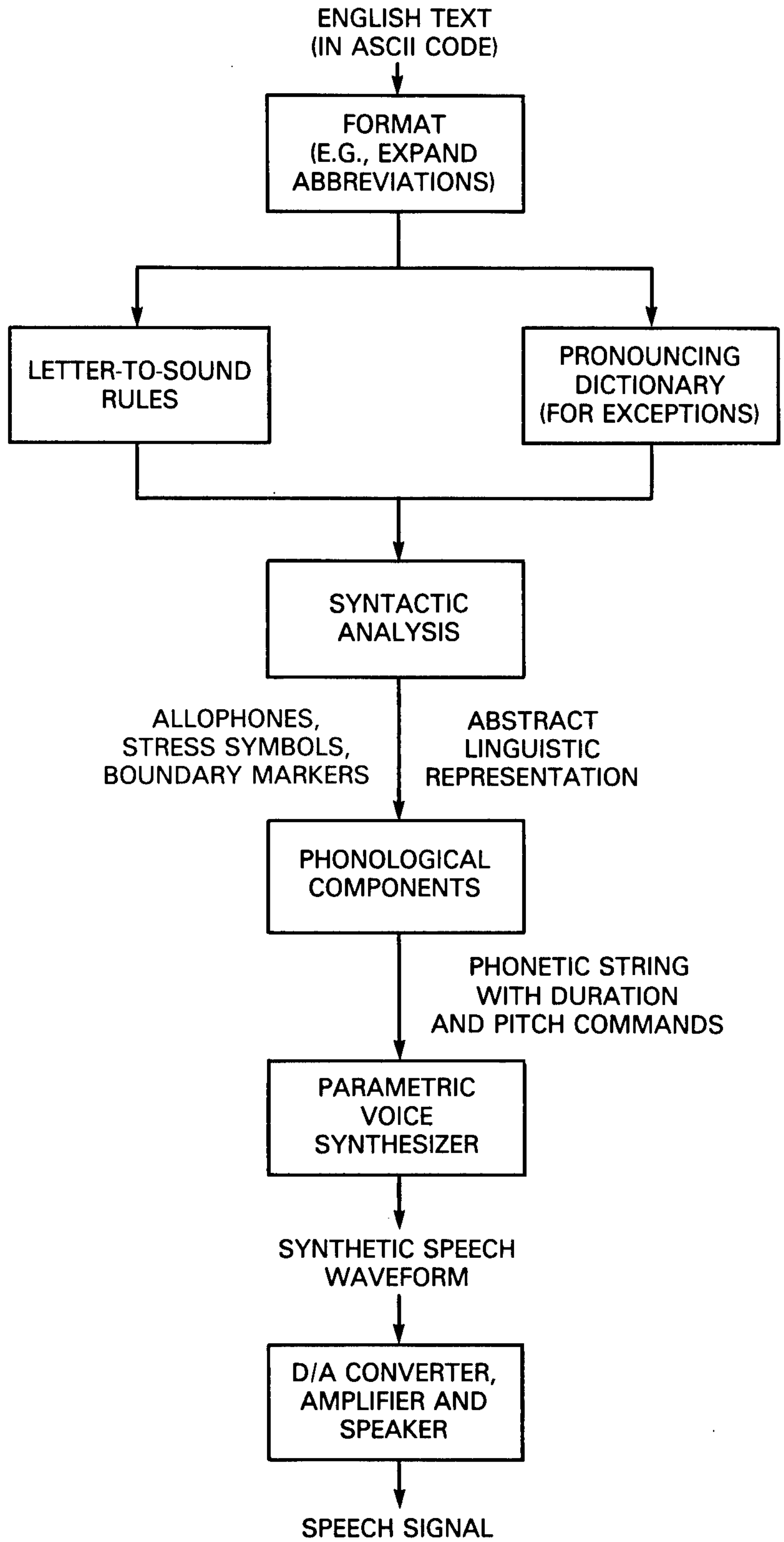


Figure V-4. Speech Quality Versus Bit Rate for Various Coding Schemes.



After Zue (1982)

Figure V-5. Text to Speech Conversion.

Programming advanced speech synthesizers, to be used with speech generation from text, is an enormous task. The flow diagram for such a state of the art system is given in Figure V-5. First, the printed text must be converted into phonemes by using a combination of rules and a stored pronouncing dictionary, taking into account pitch, intensity, and duration associated with emphasis, as influenced by word use determined by the syntax of the sentence. The resultant allophones (phonemic variations) are then fed to a phonemic voice synthesizer.

The major commercial application thus far for speech generation from text is reading systems for the blind. These products input text using optical character recognition, and output speech using a text-to-speech synthesizer. Other applications include electronic mail-to-voice, and proofreading.

K. Some Available Commercial Systems

An indication of manufacturers and currently available commercial systems is given by Table V-2.

L. Problems and Issues

- There is a tradeoff in system design between speech quality, vocabulary size, and cost.
- Problem of how to best divide the fundamental units to be used — allophones, syllables, words. The smaller units permit very large vocabularies without excessive storage requirements, while the larger units (such as phrases) provide superior speech quality.
- Memory cost considerations tend to restrict the use of the word synthesis approach.
- As the synthesizer techniques improve, it may be that errors due to low sampling rates, and inadequate consideration of coarticulation and prosodic (speech stress) effects may be the limiting factors.
- Speech compression techniques are crucial to minimize memory requirements in the synthesizer.
- The high cost of generating words for synthesizer vocabularies needs to be reduced.
- Similarly, the high cost of storing words in ROM needs to be addressed.
- Updating stored vocabularies is problematical due to the need to keep the same speaker available.

M. Forecast

Though the market for voice synthesizers is still relatively small, it is estimated that it will be close to one-half billion dollars by 1985 and will reach several billion dollars by 1990. Talking devices will have a big impact on industrial operations, a major effect on learning devices, and will probably be ubiquitous throughout home and consumer products. These devices will be a boon to the handicapped, in everything from talking typewriters and appliances, and reading machines for the blind, to speech prosthetics. It is also anticipated that these devices will be found virtually everywhere in vehicles and transportation systems.

Because of their integration into single chips, the cost of stored vocabulary devices will continue to drop so that basic hardware costs of less than $10, for units having vocabularies of several hundred words, are foreseen by the end of this decade.

TABLE V-2. Some Available Commercial Synthesizer Systems

Manufacturer	Model	Cost	Type	Comments
Votrax (Troy, MI)	VSM/1	$995	Formant	A singleboard complete system incorporating a programmable memory
	SVA		Formant	Singleboard synthesizer for unlimited text to speech (no internal word storage)
	SC-02		Formant	Synthesizer chip with phoneme library
Speech Plus, Inc. (Mt. View, CA)	Prose 2000	$3500	Formant	Singleboard system achieving an unlimited vocabulary capability by using 400 rules and a 3000 word exceptions lexicon. For use with text.
	Speech 1000	$1200	LPC	Synthesizer board with up to 6 minutes stored vocab.
Texas Instruments (TI) (Dallas, TX)	TMS 5220	$5	LPC	Single chip voice synthesizer processor.
	TMS 6100	$5	LPC	Single chip voice synthesizer memory.
	Speech synthesizer for TI 99/4A Personal Computer	$100	LPC	Text to speech implemented in 99/4A.
	TM 990/306		LPC	Speech module (does not have unlimited vocabulary capability of formant systems).
National Semiconductor (Santa Clara, CA)	Digitalker MM 54104		Mozer's Waveform Digitizer	Single chip with 256 possible addressable expressions.
Centigram (Sunnyvale, CA)	GIM	$350	Formant	An SBX module using the GI250 synthesizer chip.
	SYBIL	$495	Formant	A single channel synthesizer for the IBM PC.

TABLE V-2. Some Available Commercial Synthesizer Systems (cont.)

Manufacture	Model	Cost	Type	Comments
Kurzweil Computer Products (Cambridge, MA)	Reading Machine for Blind	$30,000	Formant	Uses Speech Plus Prose 2000 synthesizer.
American Microsystems (Santa Clara, CA)	53610	*	LPC	
	53620	*	LPC	
General Instruments (Hicksville, N.Y.)	Allophone Synthesis Module		LPC	Annunciates 64 Allophones
	SP250	*	Formant	Single channel synthesizer
	SP256	*	Formant	Single channel synthesizer, with microprocessor control
Hitachi	HD 38880	*	PARCOR	Uses Partial Autocorrelation (closely related to LPC)
Nippon Electric Corp			PARCOR	
Sanyo	LC 1800	*	PARCOR	
Mitsubishi	M58817	*	PARCOR	
Matsushita (Japan)		*	LPC	
Master Specialties (Costa Mesa, CA)	1650	$500 + Vocabulary at $50/word	Word Synthesis	
Intex Micro Systems (Troy, N.Y.)	Intex-Talker		Text-to-Speech Synthesizer	Uses a text-to-phoneme algorithm and a Votrax SC-01 chip.
Motorola			CVSD	Encoder and Decoder Chips.
Phillips/Signetics	MEA 8000	*	Formant	
	MEA 10000	*	Formant	
OKI Semiconductor			ADPCM	Encoder and Decoder Chips.

*Chip prices range from $3 to $15 depending on model and quantity. Speech Plus provides custom vocabulary generation services for speech synthesizer chips at $100/word.

REFERENCES

• Andreiev, N., "Speech Synthesis: High Technology's Dark Horse in Search of New Pastures," *Control Engineering,* September 1981, pp. 95-98.

• Berney, C. L. and Harshman, C., "Voice Ware Does It Differently, *Mini-Micro Systems,* March 1982, pp. 183-193.

• Brightman, T. and Crook, S., "Exploring Practical Speech I/O," *Mini-Micro Systems,* May 1982, pp. 291-306.

• Ciarca, S., "Use ADPCM for Highly Intelligible Speech Synthesis," *Byte,* June 1983, pp. 35-49.

• Damper, R. I., "Speech Technology — Implications for Biomedical Engineering," *J. of Medical Engineering and Technology,* Vol 6 No. 4 (July/Aug 1982), pp. 135-149.

• Elphick, M. "Talking Machines Aim for Versatility," *High Technology,* Sept/Oct 1981, pp. 41-48.

• Flanagan, J. L., "Talking with Computers: Synthesis and Recognition of Speech by Machines," *IEEE Transactions on Biomedical Engineering,* Vol BME-29, No. 4, April 1982, pp. 223-232.

• Gilblom, D. L., "A High-Quality Real-Time Text-to-Speech Converter," *Electro-82 Professional Program Record, Session 11,* Boston, Mass, May 1982, Paper No. 11/2.

• Lerner, E. J., "Products That Talk," *Spectrum,* July 1982, pp. 32-37.

• Sherwood, B. A., "The Computer Speaks," *Spectrum,* August 1979, pp. 18-25.

• Zue, V. W., *Tutorial on Natural Language Interfaces: Part 2-Speech,* Menlo Park, CA: AAAI, Aug 17, 1982.

VI. PROBLEM SOLVING AND PLANNING

A. Introduction

Nilsson at SRI originally specified problem solving and planning as being one of the four fundamental application areas of AI. However, the "weak methods," employing little domain knowledge, originally used in AI for problem solving and planning, proved inadequate for complex real-world problems. Thus, in seeking solutions in this area, larger amounts of knowledge have since been utilized. The net result has been that the "Knowledge Engineering" methodology used for Expert Systems has been adapted for use in problem-solving and planning. Thus, the boundary between problem-solving and planning and expert systems has faded and it is now common to refer to all these knowledge-based activities as expert systems and are therefore covered in that volume of this series. Nevertheless, this chapter will briefly review some of the earlier less knowledge-intensive systems and several examples of recent systems.

B. Planning Defined

Most of AI applications can be considered as examples of problem-solving, which are well covered in the other AI application areas: Expert Systems, Computer Vision, Language Understanding, etc. In this chapter we will only consider planning systems. Planning can be defined for our purposes as the design process for selecting and stringing together individual actions into sequences in order to achieve desired goals.

C. Basic Planning Paradigm

Wilensky (1983) outlines the basic structure of plans from the viewpoint of common-sense problem solving and natural language understanding. A schematic for Wilensky's basic planning paradigm is given in Figure VI-1. In this paradigm, the planner recognizes from the environment that a new situation has arisen which merits a goal. The planner then retrieves from memory a plan that might be used to achieve this goal, or generates a new trial plan if no existing plan is suitable. This candidate plan is then projected forward (via simulation) to observe the outcome. This outcome is examined to see if there are any conflicts that will arise in achieving other goals if this plan is pursued. If not, this and other candidate plan outcomes are evaluated and the maximum-valued plan is chosen. The plan, when implemented, will modify the current state-of-affairs. This impact, together with any other changes in the environment, results in a new world model with new situations that may merit new goals, so that the cyclic process of planning continues. When candidate plans are being considered, if the candidate plan overlaps existing plans for other goals, these overlapping plans may be merged to conserve resources.

A basic problem in planning is that of conflicting goals. The causes of conflicting goals are indicated in Figure VI-2. (A preservation goal is a goal to preserve an already existing condition, or is a goal not to undo a desirable state or goal resulting from another plan.)

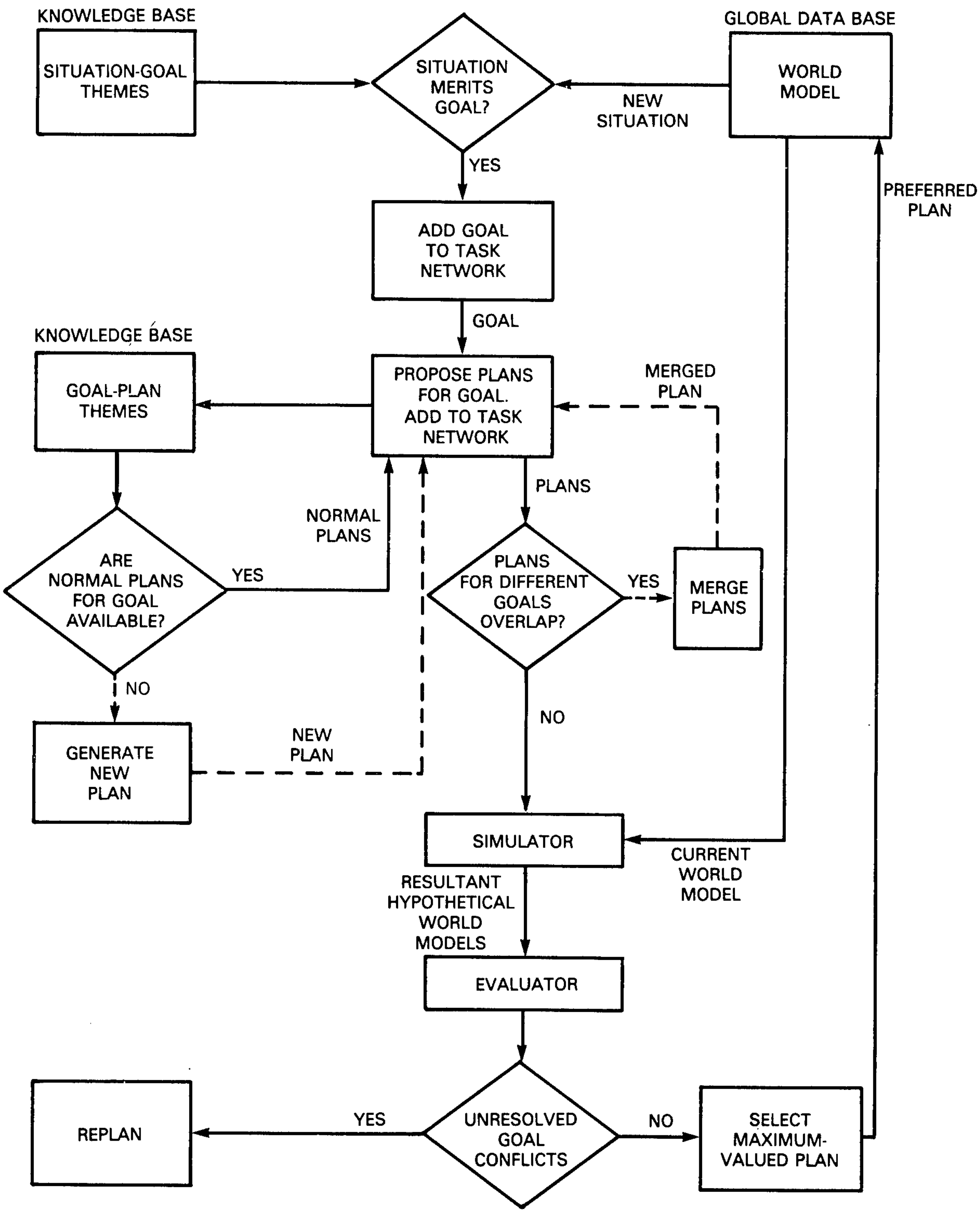


Figure VI-1. Wilensky Planning Paradigm.

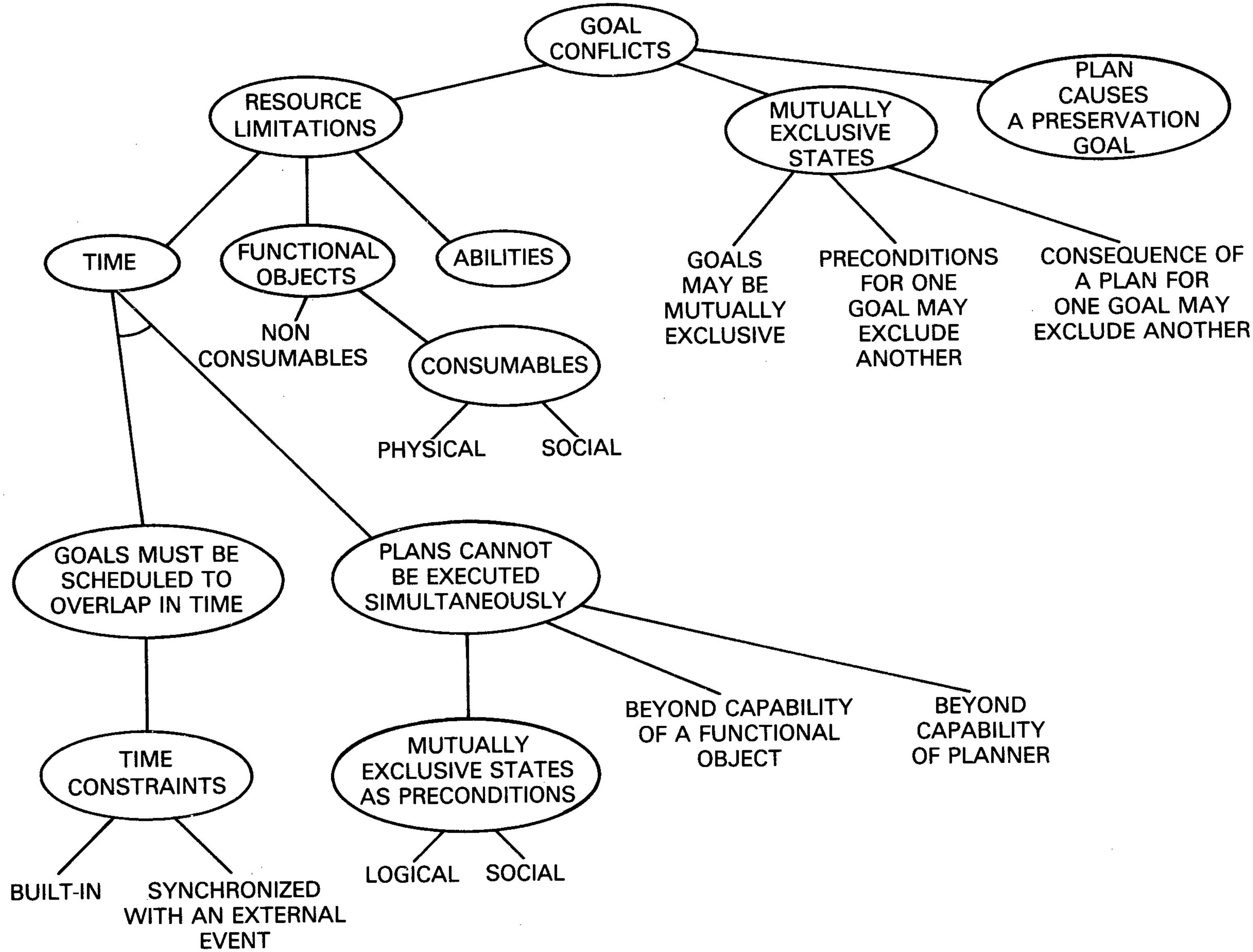


Figure VI-2. Nature of Goal Conflicts.

Problems arising from conflicting goals are dealt with by replanning or by eliminating the factors causing the goal conflicts. A flow diagram for resolving goal conflicts is given in Figure VI-3. If the goal conflicts cannot be completely resolved, then partial fulfillment of goals may be attempted or goals of lesser importance may have to be dropped. The global strategy is to achieve as many goals as possible, maximizing the composite value of the goals achieved, and not waste resources in achieving them.

DEVISER (Vere, 1983) is a good example of a planning program designed to deal with conflicting goals resulting from resource and time constraints.

Wilensky also discusses "competing goals" that arise in competitive situations. The planning strategies given in this case are to:

1) Avoid conflicts
2) Outdo an opponent
3) Hinder an opponent
4) Induce alterations in competitive plans.

D. Paradigms for Generating Plans

The major issue in any planning system is reducing search. The other key issue is how to handle interacting subproblems. The following paradigms are different approaches to addressing these issues.

Cohen and Feigenbaum (1982) discuss four distinct approaches to planning: nonhierarchical, hierarchical, script-based (skeletal) and opportunistic. Virtually all plans, both hierarchical and nonhierarchical, have hierarchical subgoal structures. That is, each goal can be expanded into several subgoals, which themselves can be further expanded, etc. until the bottom level consists of operators needed to achieve the lowest level goals. The distinction between hierarchical and nonhierarchical planners is that ". . . a hierarchical planner generates a hierarchy of representations of a plan in which the highest is a simplification, or abstraction of the plan and the lowest is a detailed plan, sufficient to solve the problem. In contrast, nonhierarchical planners have only one representation of a plan." (pp. 516-517)

1. Nonhierarchical Planning

Nonhierarchical planning does not initially distinguish between important and unimportant actions so that everything is considered in the initial plan, including cumbersome details. For complex problems, this often results in a large search. One way the search can be greatly reduced is by initially assuming subgoals independent and then trying to repair the plan to account for the interactions (as in HACKER, Table VI-1-2).

A knowledge based approach used in ISIS-II (Fox et al., 1982) is to prune the search space prior to search by using constraints, and then narrow the space actually searched by using a "beam search" approach.

2. Hierarchical Planning

In this approach, first a high level plan is formulated considering only the important aspects, then the vague parts of the plan are refined into more detailed subplans. By ignoring the details at

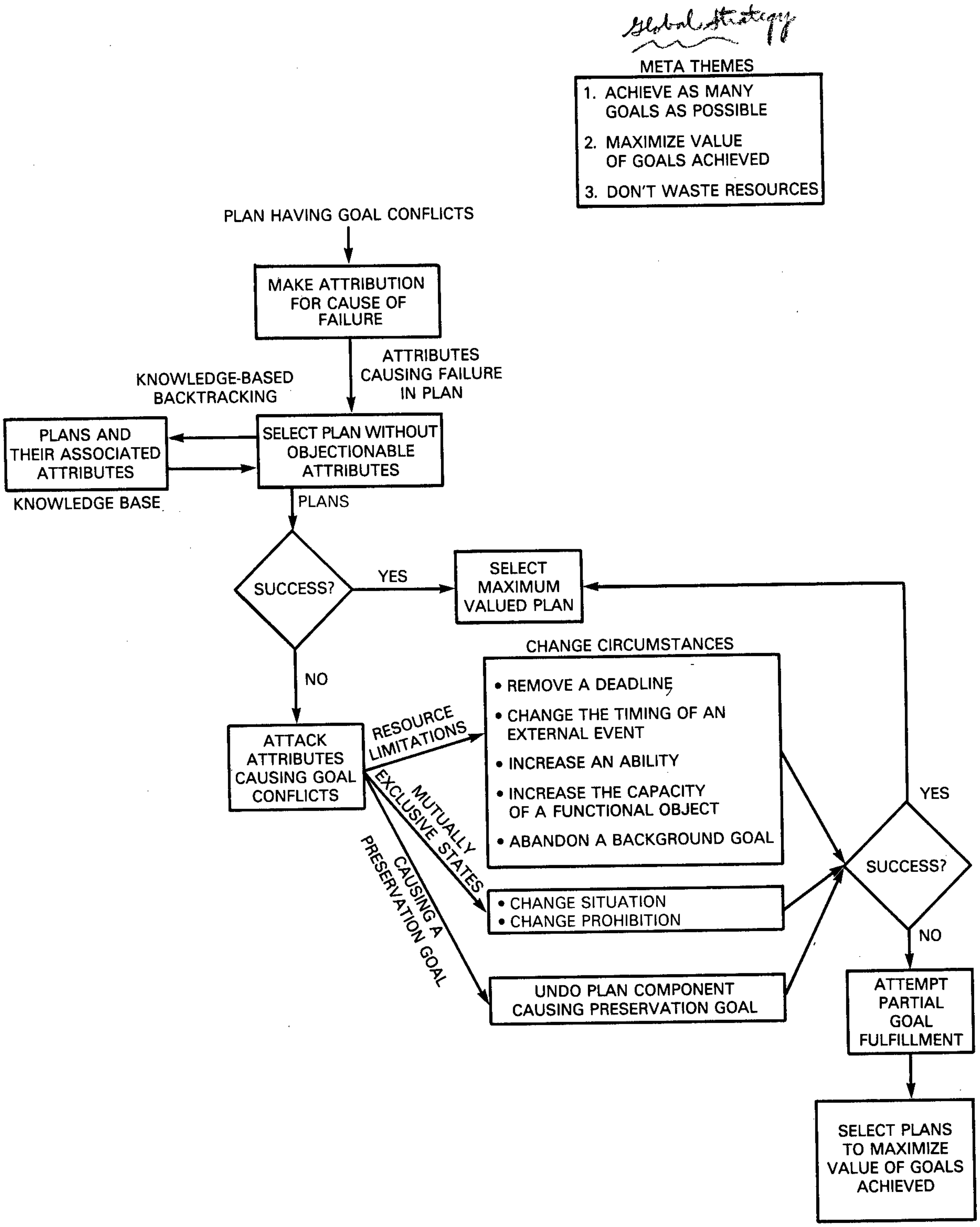


Figure VI-3. Resolving Conflicting Goals by Replanning (and/or Attacking Factors Causing Conflicts).

the higher levels, search is vastly reduced. ABSTRIPS (Table VI-1-5) is illustrative of this approach.

3. Utilization of Skeleton Plans

This approach utilizes stored plans which contain the outlines for solving many different kinds of problems. The skeleton plans are then filled in for the particular problem being solved. This technique has similarities to Schank's script-based approach to language understanding. KNOBS (Engelman et al., 1980, Table VI-1-10), a frame-based planning system for tactical air strikes, is an example of a skeletal plan approach.

4. Opportunistic Planning

Opportunistic planning (Hayes Roth and Hayes Roth, 1978) is based on the way that humans often approach planning. In this approach, the plan is developed piecewise, with parts of the plan being developed separately, and then added to, enlarged and linked together as opportunities present themselves. Planning of this sort incorporates both top-down and bottom-up components.

E. Planners

In this section we summarize the characteristics of some of the key AI planning systems that have evolved over the years. Figure VI-4 diagrams the various systems that are reviewed and their relation to the basic paradigms. Tables VI-1 outline the systems shown in Figure VI-4, using the Expert Systems format (Figure I-1) developed in Chapter I. Note that planners evolve by building on past techniques. For example, DEVISER (Table VI-1-9), the first planner to deal explicitly with time, is based on NOAH (Table VI-1-4), with facilities having been added to keep track of event "windows" and durations. Figure VI-5 presents a simplified flow chart of Deviser's core planning component.

Information on current research in planning is given in Robinson (1983).

F. Trends

Automatic Planning is still a difficult task. The current trend is toward the use of knowledge engineering to configure planners as expert systems. Thus, knowledge-based planners are included, and further discussed, in the volume on expert systems.

Another trend is toward increased concern with spatial-temporal planning. This is exemplified by Malik and Binford (1983), Allen and Koomen (1983) and Brooks (1983).

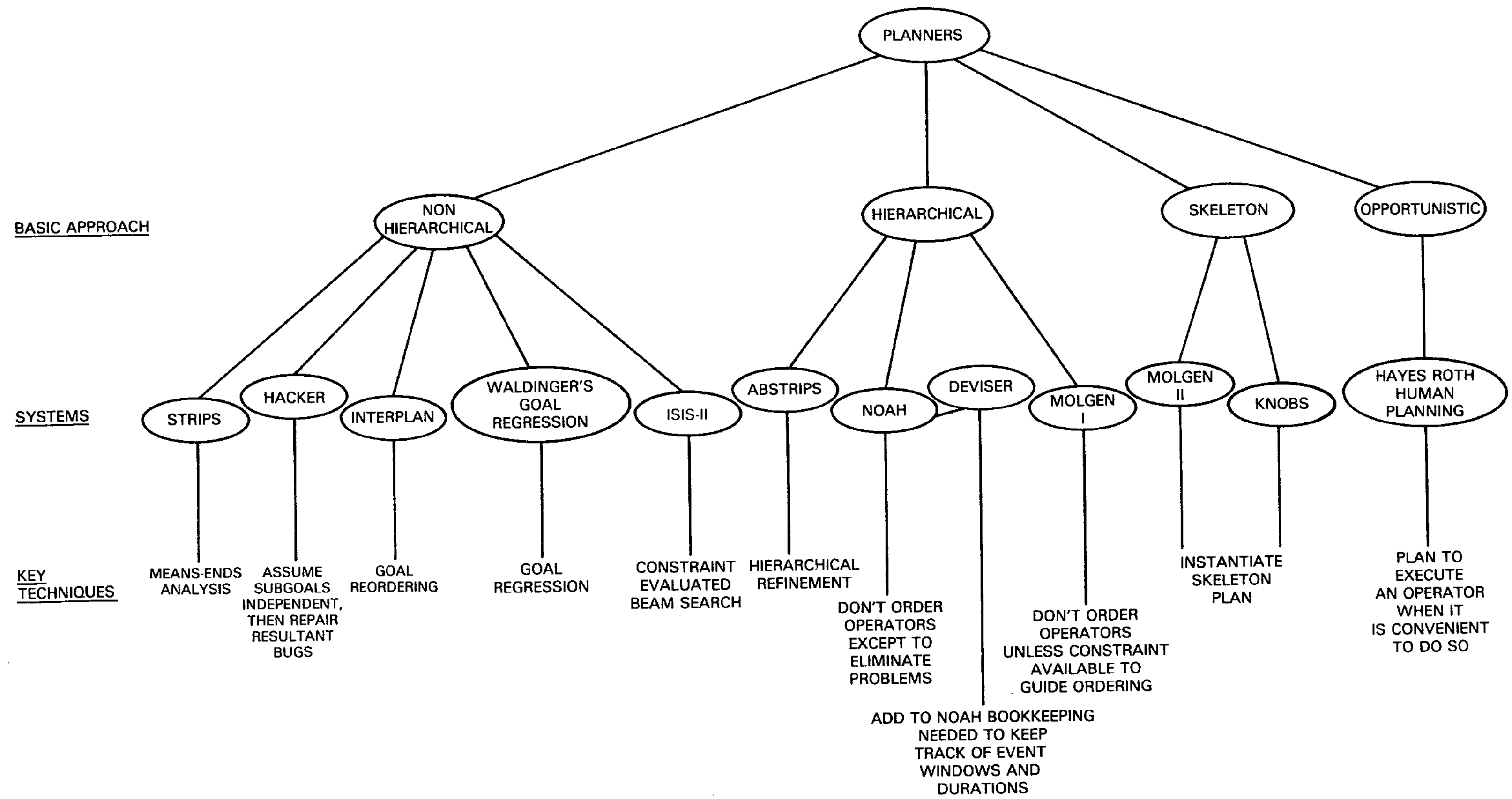


Figure VI-4. Planning Techniques.

TABLE VI-1-1. Planners.

SYSTEM: STRIPS
INSTITUTION: SRI
AUTHORS: Fikes, R.E. and Nilsson, N.J. (1971)

Purpose	Approach	Key Elements of: Knowledge Base	Key Elements of: Global Data Base (System Status)	Key Elements of: Control Structure
• Devises plans for a robot to move objects between rooms.	• Uses Means-Ends analysis (A particular implementation of GPS) • Learns by constructing macro-operators (by saving and generalizing plans).	• Uses a first-order logic representation of facts (world model). • List of problem solving operators, together with their necessary preconditions and the changes they make in the state (what is added and what is deleted from world model when they are applied).	• Goal • Initial state of the system. • Operators used thus far • Current state of the system.	• Means-Ends Analysis • Depth first search using backtracking as required.

TABLE VI-1-2. Planners.

SYSTEM: HACKER
INSTITUTION: M.I.T.
AUTHORS: Sussman, G.J. (1975)

Purpose	Approach	Key Elements of: Knowledge Base	Global Data Base	Control Structure
• Skill Acquisition: Devises a skill (set of procedures) to solve a problem. • e.g. plan to reorder blocks to a stack.	• Formulate plans to solve subgoals independently and then patch them up. (e.g., to correct interferences where achieving one subgoal may prevent accomplishment of another). • Solves Problems by: 1) Searching for an appropriate procedure. 2) If procedure does not achieve desired goal, reasons for failure are formalized as bugs. 3) Using library of bug correction procedures, the plan is debugged. • If no procedure is available to solve problem, a new procedure is written using the programming techniques library.	• Answer Library: problem-solving procedures • Knowledge Library: facts about the domain • Programming Techniques Library: to devise new problem-solving procedures • Library of generic bugs • Library of bug correction procedures	• Goals • Procedures used • Bugs	• Search for appropriate procedures to achieve goals and correct bugs. • Write new procedure when no appropriate procedure is found.

TABLE VI-1-3. Planners.

SYSTEM: INTERPLAN
INSTITUTION: U. of Edinburgh
AUTHORS: Tate, A. (1975)

Purpose	Approach	Key Elements of: Knowledge Base	Key Elements of: Global Data Base	Key Elements of: Control Structure
Planning in the blocks world, e.g. stacking blocks.	• Formulates plans to solve subgoals independently. If achieving one subgoal prevents accomplishment of another and it cannot be repaired with a procedure to achieve its prerequisite (as in HACKER) then it reorders its subgoals. The subgoal at which failure occurs is promoted — moved to an earlier position in the list of subgoals to be achieved.	• Facts about the domain. • Operators to achieve state changes — includes information about preconditions and what changes operators make in world model.	• Goals • Operators used thus far • Interferences noted	• Search for operators to achieve subgoals. • Correct interferences by reordering subgoals.

TABLE VI-1-4. Planners.

SYSTEM: Not Named
INSTITUTION: SRI
AUTHORS: Waldinger, R. (1977)

Purpose	Approach	Key Elements of Knowledge Base	Key Elements of Global Data Base	Key Elements of Control Structure
• Planning • e.g: stack blocks	Construct a plan by solving one conjunctive subgoal at a time. If a subgoal solution interferes with other goals already achieved, rather than reordering the conjunctive subgoals use "goal regression." That is, move the offending subgoal back over previously achieved goals until it finds a place in the plan where the goal will not violate previously achieved goals.	• Facts about the domain • Operators to achieve state changes — together with their preconditions and what changes they make in world model.	• Goals • Operators used thus far • Interferences noted • New subgoals (regressed goals)	• Search for operators to achieve subgoals. • Goal regression

TABLE VI-1-5. *Planners.*

SYSTEM: ABSTRIPS
INSTITUTION: SRI
AUTHORS: Sacerdoti, E. D. (1974)

Purpose	Approach	Key Elements of Knowledge Base	Key Elements of Global Data Base	Key Elements of Control Structure
Devises plans for a robot to move objects between rooms.	Do hierarchical planning by first devising a top level plan based on the key aspects of the problem, then successively refining it by considering less critical aspects of the problem. Recipe: 1. Fix abstraction levels for solutions (plans). 2. Problem solution proceeds top down (most abstract to most specific). 3. Complete solution at one level and then move to next level below.	Criticality assignments of elements in robot planning domain. Configuration of the rooms. Objects and their properties in the domain. Rules for decrementing criticality level. Heuristic search rules for each level.	Goal Initial state of system (criticality at maximum) Plans thus far. Current criticality level	Goal directed (backward chaining at each level). Top down refinement of plans using hierarchical abstract search spaces.

TABLE VI-1-6. Planners.

SYSTEM: NOAH
INSTITUTION: SRI
AUTHORS: Sacerdoti, E.D. (1975)

		Key Elements of		
Purpose	**Approach**	**Knowledge Base**	**Global Data Base**	**Control Structure**
Robot Planning System (assigns an ordering to operators in a plan, e.g., an assembly task)	• Hierarchical planner—develops hierarchy of subgoals by expanding goals. (Lowest level subgoals eventually expanded by problem-solving operators.) • Expands, in parallel, individual plans for interacting subgoals, but initially assigns only a partial ordering to operators. Stops when interference between the partial subgoal plans is observed, and adjusts the ordering of the operators as needed to resolve the interference. • Develops procedural nets to represent plans as they are developed.	• Rules for recognizing interference between plans. • Rules for resolving interferences. • Domain Knowledge —functions that expand goals into subgoals —operators to transform one state to another. Effects of actions are represented explicitly (via add lists and delete lists)	• World Model • Goal • Subgoals • Partial ordering of operators in subgoal plans. • Interference between plans.	• Least commitment. • Backward chaining.

TABLE VI-1-7. Planners.

SYSTEM: MOLGEN I
INSTITUTION: Stanford U.
AUTHORS: Stefik, M.J. (1980)

Purpose	Approach	Key Elements of: Knowledge Base	Key Elements of: Global Data Base	Control Structure
• Assist molecular geneticists in planning experiments.	• Hierarchical planner using three levels of control —Strategy space: switches between least commitment and heuristic guessing —design space: makes decisions about how plan is to develop (produces goals and constraints). —planning space: contains a hierarchy of operations. Initially plan experiments with abstract operations (merging, amplifying, reacting and sorting) and general objects (gene, organism and plasmet). As specific operators or objects are chosen to replace the abstract ones, constraints are introduced into the plan.	• Explicit meta-level problem-solving operators to reason with constraints. • Problem-solving rules. • Rules for guessing • Rules for discovering interactions between subproblems via constraint propagation. • Domain knowledge	• Goals • Partial solutions • History of guesses and their effects. • Constraints.	Constraint propagation. Least commitment Heuristic guessing Relevant backtracking Use of meta-rules to reason with constraints. Hierarchical refinement. Difference reduction.

TABLE VI-1-7. Planners. (cont.)

Purpose	Approach	Key Elements of Knowledge Base	Key Elements of Global Data Base	Key Elements of Control Structure
	• Represent interactions between subproblems as constraints. • Formulate constraints as goals to be solved. • Use constraint propagation to reveal interactions between subproblems. • Suspend problem-solving as necessary, until sufficient information is derived from the interchange of constraints (least commitment, opportunistic expansion). • Use heuristic guessing to make choices when there is otherwise no compelling reason to do so. • Retract guesses as necessary when an unresolvable problem is encountered.			

TABLE VI-1-8. Planners.

SYSTEM: MOLGEN II
INSTITUTION: Stanford U.
AUTHORS: Friedland, P.E. (1979)

Purpose	Approach	Key Elements of Knowledge Base	Key Elements of Global Data Base	Key Elements of Control Structure
Plan molecular genetic experiments	• Start with skeletal plan • Instantiate each of plan steps by a method that will work within the environment of the particular problem. • Plan steps are established by choosing techniques that (in order of priority) satisfy the criteria: 1. It will carry out the specific goal of the step. 2. Be successfully applied to the given molecule. 3. Of the techniques satisfying criteria 1 and 2, it is the best (e.g., with respect to reliability, convenience accuracy, cost, time required).	• Well organized expert domain knowledge—represented using the UNITS package: —skeletal plans classified according to utility. —objective knowledge. —hierarchical organization of about 400 techniques needed to instantiate plans.	• Goal • Skeletal plan chosen • Plan thus far.	• Proceed linearly thru plan matching skeletal steps to techniques by name, synonym, or function, and choosing the most desirable of those that match.

TABLE VI-1-9. Planners.

SYSTEM: DEVISER
INSTITUTION: JPL
AUTHORS: Vere, S. (1983)

Purpose	Approach	Key Elements of: Knowledge Base	Global Data Base	Control Structure
General Purpose Automated Planner/ Scheduler to generate parallel plans to achieve goals with time constraints. (e.g., Scheduling spacecraft actions during a planetary flyby).	• Backward chaining from unordered subgoals by: 1) Satisfying goals where possible, by linking goal nodes with the same already achieved nodes. 2) If subgoals cannot be met by linking, nodes are expanded in parallel, step by step, into activities which achieve the subgoals. 3) When two parallel expansions produce contradictions, conflicts are resolved by ordering nodes (formerly unordered). 4) If conflicts can't be resolved by ordering, DEVISER backtracks to the last choice point and tries another alternative.	• Rules for recognizing interferences between subgoal expansions. • Rules for reordering subgoal plans to resolve conflicts. • Domain Knowledge: —Operators to transform one state to another. Effects of actions are represented explicitly by add lists and delete lists. —Goal windows and durations —Event schedules	• World Model • Subgoals • Ordering of operators in subgoal plans. • Interferences between subgoal plans. • Node expansion histories. • Current windows.	• Least commitment • Backward chaining • Dynamic maintenance of windows of activities and goals to preserve consistency.

TABLE VI-1-9. Planners. (cont.)

		Key Elements of		
Purpose	**Approach**	**Knowledge Base**	**Global Data Base**	**Control Structure**
	• A start window for each activity in the plan is updated dynamically during plan generation, in order to maintain consistency with the windows and durations of adjacent goals and activities.			

TABLE VI-1-10. Planners.

SYSTEM: KNOBS
INSTITUTION: MITRE
AUTHORS: Engelman, C. et al. (1980)

Purpose	Approach	Knowledge Base	Key Elements of Global Data Base	Control Structure
Planning Consultant for A.F. Tactical Missions. Other domains include: —Naval "show of flag" missions. —Scheduling of crew activities for the NASA space shuttle.	• Assist a user by interactively accepting mission data and using it to instantiate a stereotypical solution to user's problem—checking input for inconsistencies and oversights. • Represent the stereotypical missions as frames. The checks are constraints among the possible slot values in such frames. • Uses a natural language interface (APE II).	• Targets stored hierarchically in frames —individual targets inherit from generic targets. • Frames representing protypical missions and sub-missions. • Resource data —frames representing static descriptions of object attributes, with inheritances via linkage to more generic frames. • Scripts composed of causally linked chains. • The overall knowledge base network consists of several thousand frames. • Rules for instantiation of frames and slots.	• Target • Airbase from which to fly mission. • Type of Aircraft • Armaments • etc.	• Frame instantiation uses rules and constraints. • Backward chaining of production rules in a MYCIN-like deductive manner to manage such generic choices as aircraft, weapons, support, and electronic countermeasures. • Inference mechanism uses a syntactic pattern matcher with provisions for restrictions on variable instantiations.

TABLE VI-1-11. *Planners.*

SYSTEM: ISIS-II
INSTITUTION: CMU
AUTHORS: Fox, Allen & Strom (1982)

Purpose	Approach	Key Elements of: Knowledge Base	Key Elements of: Global Data Base	Key Elements of: Control Structure
Job-Shop Planning/ Scheduling of Parts Production	• Generate schedules by heuristic search using evaluation functions based on constraints associated with costs, process applicability, machine availability, and supervisor preferences. • Set up mechanism to dynamically relax constraints as required. **Sequence for Generating a Schedule** 1. Use constraints to perform a rule-based pre-search analysis to bound search. 2. Do a constraint-directed "beam-search" where only the top-rated "n" partial paths are saved. 3. Perform post-search analysis to determine if search was effective.	• Constraints (and their importance) —Organization goals (associated with profit). —Physical constraints —Gating constraints (preconditions for object applicability or process initiation).	• Preference constraints —queue positions —machine preferences • Partial paths and their evaluations. • Work in progress • Shop status • Goals, due dates, and attributes of parts to be manufactured.	• Pre-search pruning of search space—based on constraints • Beam Search using evaluation functions based on constraints

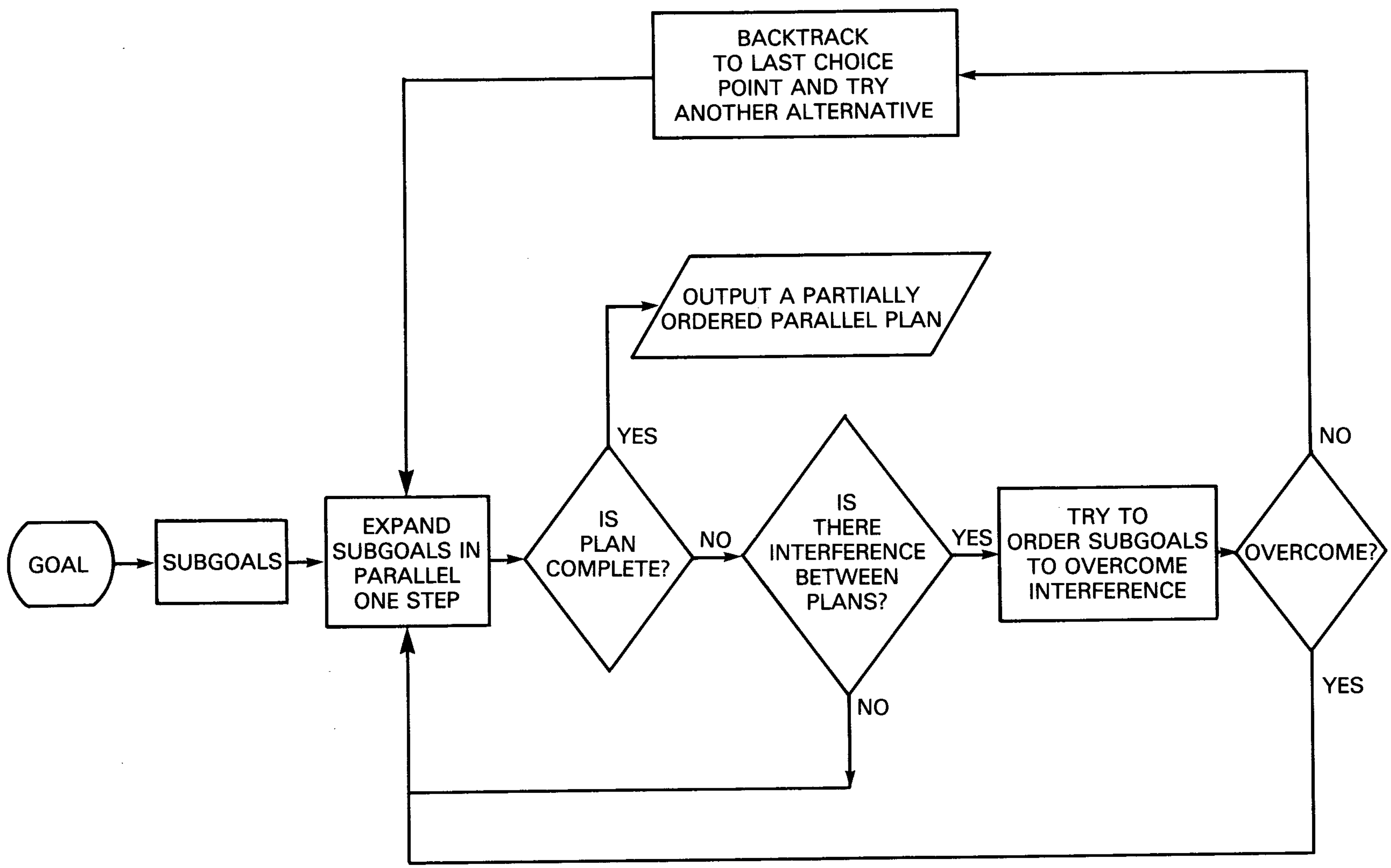


Figure VI-5. Simplified Flow Chart of Deviser's Core Planner.

REFERENCES

• Allen, J. F. and Koomen, J. A., "Planning Using a Temporal World Model," *Proc. of the Eighth International Joint Conference on AI,* Karlsruhe, W. Germany, Aug. 8-12, 1983, Los Altos, CA; W. Kaufmann, 1983, pp 741-747.

• Brooks, R. A., "Find-Path for a PUMA-Class Robot," *Proc. of the National Conference on AI,* Wash, D.C., August 22-26, 1983, Los Altos, CA: W. Kaufmann, 1983, pp 40-44.

• Cohen, P.R. and Feigenbaum, E.A., *The Handbook of Artificial Intelligence,* Vol. III, Los Altos, CA: W. Kaufmann, 1982.

• Engelman, C., Scarl, E. and Berg, C., "Interactive Frame Instantiation," in *Proc. of the First Annual Conference on Artificial Intelligence,* Stanford, 1980.

• Fikes, R.E. and Nilsson, N.J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," *Artificial Intelligence,* Vol. 2, 1971, pp 189-208.

• Fox, M.S., Allen, B. and Strohm, G., "Job-Shop Scheduling: An Investigation in Constraint-Directed Reasoning," *Proc. Second Annual Nat. Conf. on AI,* Carnegie-Mellon U., August 1982, pp 155-158.

• Friedland, P.E., "Knowledge-Based Experiment Design in Molecular Genetics," Rep. No. 79-771, Computer Science Dept., Stanford U., 1979.

• Hayes Roth, B. and Hayes Roth, F., "Cognitive Processes in Planning," Rep. No. R-2366-ONR, Rand Corp., Santa Monica, CA, 1978.

• Malik, J. and Binford, T. O., "Reasoning in Time and Space," *Proc. of the Eighth International Joint Conference on AI,* Karlsruhe, W. Germany, Aug. 8-12, 1983, Los Altos, CA: W. Kaufmann, 1983, pp 343-345.

• Robinson, A.E., "Research on Planning," *SIGART Newsletter,* No. 83, Jan. 1983, pp 27-36.

• Sacerdoti, E.D., "Planning in a Hierarchy of Abstraction Spaces," *Artificial Intelligence,* Vol. 5, 1974, pp 115-135.

• Sacerdoti, E.D., "A Structure for Plans and Behavior," Tech. Note 109, AI Center, SRI Inter., Menlo Park, CA, 1975.

• Stefik, M.J., "Planning with Constraints," Rep. No. 80-874, Computer Sci. Dept., Stanford U., 1980.

• Sussman, G.J., *A Computer Model of Skill Acquisition,* New York: American Elsevier, 1975.

• Tate, A., "Project Planning Using a Hierarchic Nonlinear Planner," Rep. No. 25, AI Res. Dept., U. of Edinburgh, 1976.

• Vere, S., "Planning in Time: Windows and Durations for Activities and Goals," *Pattern Analysis and Machine Intelligence,* Vol. PAMI-5, No. 3, May 1983, pp. 246-266.

• Waldinger, R., "Achieving Several Goals Simultaneously," In E.W. Elcock and D. Michie (Eds.) *Machine Intelligence 8,* New York: Halstead/Wiley, 1977.

• Wilenksy, R., *Planning and Understanding: A Computational Approach to Human Reasoning,* Reading, MA: Addison-Wesley, 1983.

1. Report No. NASA TM-85838	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN OVERVIEW OF ARTIFICIAL INTELLIGENCE AND ROBOTICS: VOLUME I - ARTIFICIAL INTELLIGENCE PART B - APPLICATIONS		5. Report Date October 1983
		6. Performing Organization Code RTC-6
7. Author(s) William B. Gevarter		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code RTC-6
15. Supplementary Notes		

16. Abstract

Artificial Intelligence (AI) is an emerging technology that has recently attracted considerable attention. Many applications are now under development. This report, Part B of a three part report on AI, presents overviews of the key application areas: Expert Systems, Computer Vision, Natural Language Processing, Speech Interfaces, and Problem Solving and Planning. The basic approaches to such systems, the state-of-the-art, existing systems and future trends and expectations are covered.

It is anticipated that this report will prove useful to government and private engineering and research managers, potential users, and others who will be affected by this field as it unfolds.

17. Key Words (Suggested by Author(s)) Artificial Intelligence, Machine Intelligence, Heuristic Programming, Computers, Expert Systems, Computer Vision, Natural Language Processing, Automation, Speech Interfaces, Problem Solving and Planning, Overview		18. Distribution Statement Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 121	22. Price A06

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

NASA-Langley, 1983