ARTIFICIAL INTELLIGENCE-
A HEURISTIC SEARCH FOR COMMERCIAL
AND MANAGEMENT SCIENCE APPLICATIONS

by

Philip A. Cooper

B.S., Syracuse University
(1973)

SUBMITTED TO THE ALFRED P.
SLOAN SCHOOL OF MANAGEMENT
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE IN MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1984

Signature of Author: ______________________

Alfred P. Sloan School of Management
May 4, 1984

Certified by: ______________________

John Henderson
Thesis Supervisor

Accepted by ______________________

Alan F. White
Director of Executive Education

ARTIFICIAL INTELLIGENCE- A HEURISTIC SEARCH FOR COMMERCIAL AND MANAGEMENT SCIENCE APPLICATIONS

By

PHILIP A. COOPER

Submitted to the Alfred P. Sloan School of Management on May 4, 1983 in partial fulfillment of the requirements for the Degree of Master of Science in Management

ABSTRACT

This thesis seeks to examine the rapidly growing and influential area of computer science called Artificial Intelligence; with a view towards providing a perspective on the field's:

- Historical context
- Anthropology and morphology
- What may we reasonably expect it to do

A businessman's perspective is taken throughout the thesis. The underlying question are: Is the techonology ready to be commercialized and what will the criteria be for successful products. Key issues in Artificial Intelligence are defined and discussed. Prospective product areas are identified, and desireable system attributes are put forth. Finally, moral and ethical question are examined.

Thesis Supervisor: John Henderson
Title: Associate Professor of Management Science

Acknowledgements

The author would like to express his appreciation to the following people for their special contributions which made this epistle possible. The end product is mine and I am responsible for its content. Expressing my appreciation to these kind people does not imply their endorsement. But I am in the debt of-

Zenon Zannetos, without whose help, support and encouragment I would not have come to MIT.

John Henderson, more than a supervisor, who rose beyond the call of duty to read the drafts and make suggestions under impossible time pressure.

Tom Malone for guiding me through some of the more arcane alleyways of AI.

David Hathaway, for encouraging and supporting my ideas spiritually and materially.

Peggy Cooper, for correcting my endless drafts and tolerating me during the darkest days.

Of all those arts on which the wise excell, Nature's chief masterpiece is writing well.

-John Sheffield,
Duke of Buckingham and Normandy,
Essay on Poetry

Table of Contents

Chapter 1 Prolegomenon 9

1.1 Concept and Controversy 14
1.2 History 20

1.2.1 Fact and Fancy 20
1.2.2 Philosophy 24
1.2.3 Mathematics 33

1.3 Theosophy 44

1.3.1 Cybernetics 44
1.3.2 The Church-Turing Thesis 46

1.4 The Evolution of the Apparatus 51

1.4.1 Complements and Catalysts 52
1.4.2 Automation 54
1.4.3 The Digital Computer 56
1.4.4 Future Developments 59

1.5 Palinode 60

Chapter 2 The Anthropology AI 64

2.1 Minds over Matter 66

2.1.1 A Game of "Telephone" 69

2.2 Digital Linguistics 71

2.2.1 Silicon Sign Language-The Machine Layer 71

2.2.1.1 Mathematics 74
2.2.1.2 Logic For The Logical 75

2.2.2 Operating Systems 77
2.2.3 The Compiler 77

2.3 Data Expression 80

2.3.1 Conventional Programming Languages 80

2.3.2 Applications Packages 84
2.3.3 Specialty Languages 85

2.4 Bounds of the Paradigm 87

Chapter 3 The Morphology Of Artificial Intelligence 92

3.1 The First Tentative Steps 93
3.2 Meta-knowledge and Sensory-Cognitive Problems 95
3.3 Applications in sight, but out of range 97
3.4 Knowledge Representation Today 100

3.4.1 Generality 101
3.4.2 Efficiency 103
3.4.3 Unambiguity 104

3.5 Paradigms of AI-Logic for the Illogical 105

3.5.1 The Predicate Calculus 106
3.5.2 Production Rules 107
3.5.3 Semantic Networks 108
3.5.4 Frames-Object Representation 109
3.5.5 Dealing with Uncertainty 112

3.5.5.1 Approximate Implication 112
3.5.5.2 Fuzzy Logic 113

3.6 The Inference Engine 114

3.6.1 Reasoning Mechanisms-Recap 114

3.6.1.1 Forward Chaining 117
3.6.1.2 Backward Chaining 118

3.7 System Building Tools 119

3.7.1 LISP and the LIKE 121
3.7.2 Intermediate Tools 122

3.7.2.1 Integration 123

3.7.3 The Next Generation-System Generators 123

3.8 Communications 125

Chapter 4 Artificial Intelligence Taxonomy 127

4.1 Natural Langauge[175] 128

4.1.1 The Need 128
4.1.2 Expectations 131
4.1.3 Requirements for True Functionality 134
4.1.4 Constraints on Natural Language 136

4.2 Other User Interfaces 138
4.3 Expert and Knowledged Based Systems 140

4.3.1 Architecture 142

4.3.1.1 Human Expert 145
4.3.1.2 Explanation Systems 146
4.3.1.3 Inference Engine 146
4.3.1.4 User Interfaces 146

4.3.2 Construction Process 147

4.3.2.1 Knowledge Engineer 149
4.3.2.2 Knowledge Acquisition Systems 150

4.3.3 Future Enhancements 151

4.3.3.1 Learning Subsystems 153
4.3.3.2 More Powerful Inference Engines 153
4.3.3.3 Natural Language 154

Chapter 5 Prospects For Commercialization 155

5.1 Natural vs Artificial Intelligence 156

5.1.1 Appropriate Tasks 158

5.2 An Expert System Prescription 161

5.2.1 System Attributes 162

5.2.1.1 WAG (sic) 162
5.2.1.2 Normal Returns File 163
5.2.1.3 Alternative Methodologies 164
5.2.1.4 Meta-Knowledge (again) 164
5.2.1.5 Cross Disciplinary Knowledge 165
5.2.1.6 Learning 165

---

175. The terms "natural language" and "English" are used interchangably. The same issues apply for "foreign" languages as well

5.2.2 User Interface 166

5.3 Generic Product Categories 167
5.4 Commercial Target Markets 167

Chapter 6 Quo Vado? A Look Ahead To Adolescence 170

6.1 Man, Golem, Responsibility and Ethics 171

6.1.1 Modern Times 173
6.1.2 Guarding Against HAL 176

Chapter 7 Bibliography 185

Appendix A Examples of Expert Systems 189

Appendix B Worldwide Artificial Intelligence Activity[176] 196

Appendix C Expert Systems-Development Flow Charts (Advanced) 200

Chapter 8 References 209

---

176. Ibid., pp. 251-253

Chapter 1

Prolegomenon

Question: You seem to be saying that AI[1] programs will be virtually identical to people, then. Won't there be any differences?

Speculation: Probably the differences between AI programs and people will be larger than the differences between most people. It is almost impossible to imagine that the "body" in which an AI program is housed would not affect it deeply. So unless it had an amazingly faithful replica of a human body-and why should it?-it would probably have enormously different perspectives on what is important, what is interesting, etc. Wittgenstein once made the amusing comment, "If a Lion could speak, we would not understand him." It makes me think of Rousseau's painting of the gentle lion and the sleeping gypsy on the moonlit desert. But how does Wittgenstein know? My guess is that any AI program would, if comprehensible to us, seem pretty alien. For that reason, we will have a hard time deciding when and if we really are dealing with an AI program, or just a "weird" program.

Question: Will we understand what intelligence and consciousness and free will and "I" are when we have made an intelligent program?

Speculation: Sort of- it all depends on what you mean by "understand". On a gut level, each of us probably has about as good an understanding as is possible of those things, to start with. It is like listening to music. Do you really understand Bach because you have taken him apart? Or do you understand it that time you felt the exhilaration in every nerve in you body? Do we understand how the speed of light is constant in

1. Artificial Intelligence

every inertial reference frame? We can do math, but no one in the world has truly relativistic intuition. And probably no one will ever understand the mysteries of intelligence and consciousness in an intuitive way. Each of us can understand **people**, and that is probably as close as you can come.

-Douglas R. Hofstadter,
Godel, Escher, Bach[2]

The reasoning animal has finally made the reasoning machine.

Who dares feign surprise at the inevitable? It's human to exhibit intelligence, and human to make machines. The combination, not to say the collision, of the two is the most human of stories.

-Edward A. Feigenbaum, Pamela McCorduck,
The Fifth Generation[3]

Everything that can be thought at all can be thought clearly. Everything that can be said at all can be said clearly. But not everything that can be thought can be said.

-Ludwig Wittgenstein

Machines, mechanical or electronic, are indispensable to our way of life today. Only computers make possible the advanced communications, accounting, financial, fulfillment and other services deemed so necessary to society. Computers are needed to design computers, keep us productive and competitive in this world, and explore space to find the next world.

While one may argue that these are mixed blessings, it is easy to see that we would be unwilling to give them up. The general standard of living is higher than at any time in history and the conveniences provided by devices with a "chip" imbedded somewhere are as seductive as they are addictive. As long as machines have been altering societies, the desire to make them do more has exceeded the state of the creator's art. It is a curious phenomenon, but it appears that a modification of Parkinson's Law[4] can be applied to many types of technologies: For a given technology, the expectations of the users or beneficiaries of that technology, not themselves trained in it, will exceed the current capabilities of that technology by at least an order of magnitude. Thus expectations don't merely **rise** to meet capability, they lead it. People **expect** computers to double in power every year, the space shuttle is expected to outdo itself on every mission and even minor failures are seen as major shortcomings and setbacks. Nowhere is this level of expectations higher than it is in the computer field.

Computers have always fascinated the population. The promise of extending human capabilities through these machines early on led people to extend the thought to making the machines act like us as well as for us.

The exponential growth of the power[5] of these machines has encouraged popularizers, often aided and abetted by researchers and computer company marketing departments, to make ever increasing claims for the capabilities of the technology. Paradoxically, as each generation of machines has had more power, they have become harder to use, to their full potential, without special expertise or special high level software to enable humans to more easily harness their power. It is hard to imagine Mr. and Mrs. America at the console of the Cray II or Cyber 205 improving their personal productivity. The way the mainstream of computer science and commercial computer vendors are addressing this issue is by making their products more "user friendly".

The notion of making machines in general, and computers in specific, "friendly" and accessible to the user[6] is relatively new and is particularly a phenomenon of the computer age. It was historically more important that the machines were programmed ("hardwired", if you prefer) to do the task right than it was for the machine to be able to easily communicate to its operator. Little "judgement" was required by either the machine or the

---

5. Power meaning processor rates "MIPS" (millions of instructions per second), available inexpensive "RAM" memory, high speed disks, more efficient software, parallel processors, etc. It may be useful to think of power as "horsepower".

6. As a matter of convenience, the use of the words "user" and "end user" in this paper is meant to mean the unsophisticated user. Such users may be very knowledgeable in their area of expertise; the unsophisticated reference is to expertise in the operation of the computer, per se.

operator. No matter how complex the tapestry being woven, the Jacquard loom followed the same pattern. Nothing the attendant could do would alter that pattern. Only the (systems) expert could configure and insert new control cards (programs).

However our seventeenth century French mill worker probably considered the automated loom "intelligent". It could weave the most complex patterns with utter reliability and repeatability without human intervention. He had the precedent of even more fantastic machines and a rich lore of automata to re-enforce his perception of intelligence. Of course, his education was limited or non-existent. So he probably did not contemplate machine intelligence at great lengths. But these machines weren't intelligent in the modern sense. Operators could attend many machines, but their task was really to keep raw materials supplied and fix jams. They didn't run the machines as much as service them. This type of man-machine interface has evolved very little for many applications. Today, no matter how user friendly one might view the cash machine at the bank, it is unlikely[7] that many would call it intelligent. It's responses are bounded, completely predictable, and exhibit no heuristic judgement. Yet surely the cash machine would have been considered intelligent in the context of ENIAC, definitely in comparison with the Jacquard loom. This, of course, reflects that society's baseline view of a "dumb" machine keeps rising, commeasurate with the population's education and exposure to, and

---

7. Barring failures.

de-mystification of, technology.

> The can isn't as user friendly as it used to be,
>
> -Director of Marketing Research, Campbell's Soup,
> Wall Street Journal, 3/28/84

1.1 Concept and Controversy

The concept of Artificial Intelligence [AI][8] is quite distinct from mere user friendliness. Something user friendly is easy to use. The concept in that form may be applied to a vacuum cleaner. Something intelligent is quite another matter. It may or may not be easy to use, this is a matter of implementation. Indeed, one can have an artificially intelligent system which is not user friendly.

In the debate about the promise and limitations of Artificial Intelligence the operative word is **Artificial**, the controversial element in the **Intelligence.** Everyone agrees that a machine is, by definition, artificial because it is not "alive". But whether man-made physical object whose essence is the inert element silicon, can be made intelligent, is a subject whose surface conceals its depths. It is also, as the previous quotes from Hofstadter and Feigenbaum illustrate, (See page 10) a very controversial

---

8. The field in general, tends to use the terms "Artificial Intelligence", "expert systems", and "knowledge based systems" interchangeably; although, strictly speaking, they are not. We will follow the field.

notion in computer science. It is even a more controversial notion in a society steeped in 1984.

Given the same information about a situation an intelligent device might conclude differently than you or I. Whether the machine "thought" about it in the way you or I or any given individual would is central to the debate. Who is "right" is yet another matter. The machine is only reflecting the thought patterns of those who programmed it, which we may or may not agree with.

> Thou speakest wiser than thou art ware of,
>
> -Shakespeare,
> As You Like It, II, iv, 57

It is "right" only to the extent that the process it was programmed with is right. It is predictable to the extent one knows its heuristics or, more properly, its programmer(s') heuristics. In the same sense, people do not always agree with the "experts" on a given matter. Furthermore, the "experts" are not always right.But if it was perfectly programmed with its creator's thought process, if that were possible; then would it be thinking? Would it be predictable. Is its creator consistently predictable?

Interestingly, there is a body of thought that holds computer programs, AI or otherwise, are not always predictable, even by those who programmed them. This line of reasoning suggests that it is literally impossible to document every possible resultant that a [reasonably] complicated program might arrive at. This is why attempts to program computers to play chess based on an exhaustive search of the current possible moves and, for the purpose of strategy, possible moves by both players three moves ahead, always failed. Claude Shannon calculated[9] that a computer playing chess by such methods (which, incidentally, most people unfamiliar with AI believe **is** how computers play chess) would have to calculate about 10^{120} possible moves. A machine calculating one variation every **millionth** of a second would require 10^{95} **year** to decide on its **first** move![10] This is one reason that AI researchers have devoted so much effort to developing "grandmaster" level chess playing programs. Many elements of chess provide a useful experimental paradigm for human thought processes.

If every possible resultant is undocumentable, then all outcomes cannot be known. Thus some unpredictability my be inherent in all systems. Of course, one man's unpredictability may be another man's "bug". Furthermore some of this unpredictability may be due to quirks or bugs in the operating system, compilers, or assembler. Yet some believe that this very unpredictability means that the machine is exhibiting intelligence. Although this is a debatable definition of intelligence, it does introduce the notion of a random, or at least a hyperbolic associativeness as a factor in originality and therefore intelligence. This part of such logic is at least intuitive.

Daydreaming, or "letting your mind go", frequently leads to new thoughts and ideas, and also solutions, to what may have seemed to be

---

9. Shannon is discussed in a forthcoming section

intractable problems. However, we are often hard pressed to explain a logical, documentable, repeatable process that led to that "breakthrough" idea. Further, under different conditions, people, even the same people, may arrive at completely different approaches to the same problem. Factors such as mood, weather, or the nature of the group one is in during a "brainstorming session, or whether one is alone; can easily trigger widely different conclusions even if presented with exactly the same data. Much interesting work has gone on in game theory and statistical decision making theory looking for theories to explain how people arrive at decisions, especially when there are multiple criteria at hand. Many systems have been proposed. But to the extent they try to create uncertainty in the process or combine differing criteria in imaginative ways to generate an (a priori) unexpected alternative, they are still relying on statistics. The uncertainty level is always quantifyable. Under many simulations, things regress to a mean, or produce random garbage. Clearly there is more involved in originality and intelligent problem solving than stochastic processes.

Computer programs have been devised which purport to demonstrate intelligence by "composing" music. Can it? The words of the program's creator, Max Mathews:

> Is the computer composing? The question is best unasked, but it cannot be completely ignored. An answer is difficult to provide. The algorithms are deterministic, simple, and understandable. No complicated or hard-to-understand computations are involved; no "learning" programs are used; no random processes occur; the machine functions in a perfectly mechanical and straightforward manner. However, the result is a sequence of sound that are unplanned in fine detail by the composer, even though the over-all structure of the section is

completely and precisely specified. Thus the composer is often surprised, and pleasantly surprised, at the details of the realization of his ideas. To this extent only is the computer composing. We call the process algorithmic composition, but we immediately re-emphasize that the algorithms are transparently simple.[11]

Similar examples can be given for checkers, chess, medical diagnosis, authoring short stories[12] and many others. On one level we shall see that **apparently** intelligent is a good operational definition. People view others as having varying degrees and **types** of intelligence. The term "intelligence" is applied to Dolphins and Chimpanzees as well as people. Few would question that any of these mammals are intelligence, the question is a matter of degree, and to some extent, scope.

The issue may ultimately be reduced to a contextual, or relative, one. A human's (or mammal's for that matter) intelligence is usually measured in comparison to the complexity of the task. His "effective"[13] intelligence is what he is judged by. Some would be considered geniuses at particle physics or philosophy, others at painting or music composition. Both, perhaps, equal on an absolute scale, but not a relative one. Einstein was a very amateur violin player, Norbert Weiner was only slightly above average at chess, and history records no contributions by Rembrandt to mathematics.

---

12. There is a report of a program devised to churn out pornographic paperbacks for that apparently insatiable market. But it is unclear whether the prose, or the people reading it, would impress the dispassionate (no pun intended) observer as intelligent.

13. A term coined by Professor John Henderson of M.I.T.

There are currently expert systems which are virtually the equal of physicians at diagnosing infectious diseases and cardio-pulmonary ailments. Expert systems can diagnose wire line data for oil companies at the expert level. Examples abound. They certainly meet the contextual intelligence test. But such expert systems display no originality. None has conceived a cure for a disease, not already in pharmacology. None has invented a better way to drill, or operate on the heart. They think, but they do not know. If there is to be a significant contribution to commercial software and society through Artificial Intelligence technology, it will pivot on where the lines will be drawn between "Artificial" and "Intelligence"; "knowledge" and "knowing"; "diagnosing" and "conceiving."

The issues behind Artificial Intelligence are not simple. To understand the the framework we will be discussing commercial products in, it is necessary to have a good understanding of the history and relevant issues of the disciplines from which AI emerged. We can't do more commercially than we can do theoretically.

1.2 History

1.2.1 Fact and Fancy

Our Artificial Intelligence ancestral tree has three principle roots. Philosophy (particularly Epistemology[14] and Metaphysics), Mathematics and Automation. In a discussion of the heritage of Artificial Intelligence, the boundary between philosophy and mathematics is occasional elusive. Thus we will examine the intersections of these areas, whenever they occur, in order to better understand the history of AI.

---

14. That branch of philosophy dealing with the theory of knowledge. According to Popkin[15]: "The attempt to discover the means by which our knowledge is acquired, the extent of our knowledge, and the standards and criteria by which we can reliably judge the truth or falsity of our knowledge."

I propose to consider the question "Can machines think?"

-Alan Turing,
Computing Machinery and Intelligence

The Analytic Engine has no pretenses **whatever** to originate anything. It can do whatever **we know how to order it** to perform.[16]

-Lady Ada Lovelace,[17]
Memoir, 1842

The real problem is not whether machines think, but whether men do.

Burrhus Frederic Skinner,
Contingencies of Reinforcement, ch. 9

As long as Man[18] has been trying to relieve his drudgery and extend his reach, he has tried to imbrue machines with a trifle of his own divinity.

The mythical Joseph Golem,[19] more than an automaton but less than a human, reoccurs throughout literature, frequently with unforeseen and often disastrous consequences.[20] The idea of a machine enough like us to relieve

---

16. **Emphasis** is Lady Lovelace's

18. Used for expediency only, no offense meant to female readers.

19. Originally the name of the servant fashioned from clay by the High Rabbi Judah ben Loew (circa 1580).

20. Such as Mary Shelly's Frankenstein [1818], E.T.A. Hoffman's The Sandman [circa 1815], Offenbach's The Tales of Hoffman [1880]; and innumerable quasi-automatons in various "Star Trek" episodes.

us of the routine and repetitive is inherently very attractive. The idea of becoming subservient to a machine is equally repellant. The ploy of the machine run amok has always been fuel for science fiction mills, but the depths of the fear are evident in society's concern about computers, robots and, more broadly, about genetic engineering. Traditionally the Yang of human judgement and compassion has triumphed over the Yin of cold machine rationality; when such rationality has taken on a sinister direction. Cultural history throughout the world portends horrible fates for those who mimic the Gods or the Gods' right to bestow and set the limits to life. When facing the future, we need to be reassured that we are still the masters of our own fate.

Most people have an innate fear and loathing for that which they don't understand, can't control, are [semi] at the mercy of. This phenomenon manifests itself throughout history. The Luddities destroyed mills in late eighteenth century England in a misguided attempt to preserve their jobs. Craft guilds have always resisted automation. Even today's unions fight robotics.

> Established technology tends to persist is the face of new technology,
>
> -Gerrit A. Blaauw,
> Quoted in -The Official Rules

Science fiction has an unnerving way of predicting and then becoming reality. Computers have long been cast as the progenitors of "1984".

Articles are already being written proclaiming expert systems the replacements of managers. After decades of trying to make computers [appear] more intelligent, be more useful and increase our productivity; some commentators are now becoming concerned with the effects of accomplishing just that.[21] In fact, many studies[22] have shown that automation has always increased total employment, usually due to the rapid expansion of an existing business, such as textiles, or the creation of new businesses such as computers themselves. Automation, per se, does not increase productivity, [23] or reduce unemployment.[24]

Nevertheless if the computer has brought a new dimension to these fears, the prospect of Artificial Intelligence can propagate the worst Man vs. machine" nightmares. To many people it is only a short step from the prospect of a machine threatening the utility and comprehensiveness of a person's intelligence, [25] to questioning the need for the person at all. Who can forget "HAL" in Stanley Kubrick's classic movie 2001? The preceding is very germane to this discussion of Artificial Intelligence. The implications of the intellectual and moral questions of machine intelligence will, as they have in the past, be the central issue defining the progress made in AI research, and the acceptance of "intelligent" products that may result.

---

24. When viewed on a societal basis.

25. The "adder" is now extinct in even the most reactionary accounting firms and, more directly, computers are diagnosing some diseases as well as physicians

1.2.2 Philosophy

> There is something in this, more than natural, if philosophy could find it out.
>
> -Shakespeare,
> Hamlet, II, ii, 392

Philosophy is the story of man's quest to understand himself, his place and his purpose. One must be able to do that before one can build machines to emulate[26] thought. Changes in epistemological theories and dogmas of various times have greatly influenced directions in AI.

The philosophical heritage of AI extends back to the Greeks, but traces its modern roots from a remarkable series of philosophers starting in the sixteenth century. The modern argument was cast be Rene Descartes (1596-1650). Descartes' famous **Treatise on Man** established the "Mind vs Matter" (or Body) dualism.[27] This dichotomy holds that there is a separation between the physical world and the mind, (intelligence) each completely independent of the other. The physical world operating according to God's laws (see footnote), the mind unextended and dealing with judgement, thinking, feelings and such.

The central problem of Cartesian metaphysics is how the mind and the

---

26. Yes, "emulate" is a controversial word in the AI debate.

27. Actually, Descartes' metaphysics encompassed God, Mind and Matter. The issue, for our purposes, is in the mind-body dualism.

body are related at all. If they existed apart from one another, how did they interact? Descartes himself never dealt with the issue to his or his contemporaries' satisfaction. He cobbled up a theory that mind and body met through the pineal gland at the base of the brain though this created as many problems as it solved. Descartes knew he was hedging.

Scholars have long debated why such an apparently rational man as Descartes put such a heavy mystical emphasis on God in his philosophy. He may well have been aware of the persecution of Galileo, and decided discretion was the better part of valor. But it may also be that only an appeal to the Almighty could bridge the mind-body gap. This dichotomy has been at the heart of western philosophy ever since. Although a thorough examination of this is beyond the scope of this paper, I mention it to highlight how complicated the notion of where intelligence resides is and what its relation is to the physical world. It is a striking notion that the Cartesian dialectic is still at the heart of the debate about whether machines can think or merely do as they are told. And if they are supposed to think, how the mind (human thought) can be made to reside in the body (machine).

This dichotomy led to some brilliant thinking by philosophers and mathematicians alike examining both sides of the mind-body dialectic. Newton had completed his monumental **Principia** (1687) explaining the mechanics of the natural universe. Newton codified "God's Laws" which Descartes postulated governed the universe. (See page 24.) He evidently believed that if the natural order of things could be rationally and deterministically explained, the metaphysical (or intellectual) should be able to be so codified. In the preface to **Principia** he writes:

> "I wish we could derive the rest of the phenomena of Nature by the same kind of reasoning from mechanical principles...".[28]

If the universe, created through the power and mystery of God,[29] could be explained by Newton in mathematical terms, why couldn't man's thought, also a creation of God, be explained in an appropriate calculus? This is an especially powerful thought which has tantalized and plagued man ever since. The struggle to resolve the mind-body dichotomy runs through the next several centuries of metaphysics up to the present day. So too does the question of whether we are recursively rational to our core, or essentially mystical. It certainly lies at the heart of the Artificial Intelligence controversy.

Epistemology has always wrestled with the notion of what it is possible to know, and how we acquire knowledge. Related to the mind-body dichotomy, the issue is to what extent we can perceive and know things, if we are limited in our information gathering abilities by our senses. Philosophers who subscribe to this view are known as **Empiricists.** Some of the earliest thinking in this area was done by Plato, frequently speaking

---

29. The atheist and agnostic view being ignored, for our purposes

through the dialogues of Socrates.[30] In dialogues such as **The Meno** and **The Republic** , and through such devices as the Allegory of the Cave, Plato develops the idea that the real world is unknowable by means of information received (perceived) by the senses. This is because reality is perfect, and our senses are imperfect. The imperfect cannot know the perfect, only an illusion or approximation of it. One can only know truth through the use of pure reason, and in particular, mathematics. It is easy to see how old the parallels are to some of the most fundamental debates about machine intelligence.

Plato's thinking about the nature of senses and knowledge was refined by Descartes and Francis Bacon. Thomas Hobbes (1588-1679) added the notion that human thought is not always [formally] logical, but works in [apparently] uncoordinated associative ways. Baruch Spinoza (1632-1677) developed the notion of **parallelism** or the dual aspect of nature. This view is that the logical order of the mind was identical with the physical order of nature in that they were both aspects of the same thing. Which is to say God, Substance or Nature. The idea of Plato's true reality expressed as a "superset" entity. Every mental thought has an equivalent physical manifestation.

It was the ideas of John Locke (1632-1704), which founded the modern school of epistemology and metaphysics known as the rational empiricists. In Locke's **Essay Concerning Human Understanding,** Locke laid out what might be called the first modern explanations of heuristics as the way in which man learns and applies his knowledge. Locke believed in man's intellectual rationality. He felt that even the most complex ideas were built up from simple ones and that man's attitudes and beliefs were shaped by experiential as well as intellectual data. But his epistemology leads to a most frustrating conclusion and one that has created difficulties for all empiricists: How can one have knowledge about abstract ideas, ideas not necessarily formed through sensory input? For example, the idea of romantic love, or any new idea which has not been voiced before. If intelligence is the result of sensory input, deductions thereupon and is associative, how could such ideas come into being? Assuming there were answers to these questions, how could they be built into a machine?

Bishop George Berkeley (1685-1753) continued the empiricist tradition. Berkeley wrote: "Esse est percept"—The existence of things consists in their being perceived.[31] Like Plato and Locke before him and Hume after, Berkeley was saying that all we can know is what our senses perceive, our experiences. Taken literally, our knowledge, especially of "reality" is always imperfect.[32] We cannot know what we cannot perceive. Berkeley was a Anglican bishop, he attempted to overcome the dilemma, posed in Locke's philosophy, by suggesting there was a greater consciousness, a divine one, which held all the kinds of thoughts one could not develop from sensory input alone and made those thoughts available to men. An interesting

---

32. This line of thought is at interesting variance with the common sense notion that what we perceive **is** reality

conclusion of this thinking is that things exist because they are in God's mind and that man receives his ideas from this divine consciousness. This philosophy was the subject of a famous limerick by Ronald Knox:

> There was a young man who said, "God
> Must think it exceedingly odd,
> If he finds that this tree
> Continues to be
> When there's no one about in the Quad."
> Reply.
> "Dear Sir:
> Your astonishment's odd;
> I am always about in the Quad.
> And thats why the tree
> Will continue to be
> Since observed by
> Yours faithfully,
> God."

Georg Wilhelm Freidrich Hegel (1770-1831) took the religion out of Berkeley's notions of a divine all encompassing mind and postulated the existence of an abstract **objective or absolute** mind. The Hegelian dialectic characterizes the absolute as trying to bring the physical world to terms with itself through the use of **thesis and antithesis** as bracketing attempts to say something definite about the universe. The resolution of differences between the thesis and antithesis yields **synthesis**—a proposition incorporating the perceptions of truth of both. This resultant view becomes [our] reality. From chemical to biological, humans are merely the current physical manifestation of the evolution of the absolute itself. It is interesting to speculate who, or what, will come next. If Berkeley and Hegel's notions were literally true, we could be sure that if we are able to imbrue computers with intelligence, it certainly will be artificial!

David Hume (1711-1776) returns us to the rational empiricists philosophy. He took the epistemological quandaries of Locke's metaphysics to their logical ends. In his **Treatise on Human Nature** he reached the conclusion that it isn't possible to know anything about the universe. We are completely constrained by our senses and unique associative ways of thinking. But Hume's contribution to our story was his attempt to formulate laws of association to describe how we think and evaluate and act upon information we experience in the world.

So the empiricists have willed us the problem of how we are ever to know "truth" and reality. But in their philosophies, the ideas of associative laws, complex ideas from simple components, and heuristics as the primary method of human decision making came into our intellectual lexicon.. They have helped define how we think. They believed that man's thinking can be analyzed, even if we can't know the ultimate truth or reality necessary to support many of those who claim that machine can be made to think like us. The notion of "man as machine" continued to develop in parallel with the idea that thought could be objectively, if not mathematically analyzed. Diderot (1713-1784) took this view, as did La Mettrie, if in an odd and eccentric way, in **L'Homme Machine.** The mechanistic view grew in empiricist soil.

But not without its traditional opposition. The most prominent was Immanuel Kant (1724-1804). In **The Critique of Pure Reason** Kant argues for analytic and synthetic "a priori" knowledge. Kant held that our knowledge is a combination of our experiences, limited though they might be by the senses, and the a priori experiences inherent in the mind. This is a variant of the "nature/nurture" argument.[33] A hard combination to render in silicon.

The preceding has been, by necessity, a superficial treatment of the basic philosophic lines of thought which form the theater upon which the play of Artificial Intelligence is staged. But these philosophies are more than historical arguments in the AI debate. While the focus of this thesis is admittedly commercial, it is worth considering how far one might reasonably expect to be able to take the concept of expert/knowledge based systems and intelligent devices. There are also some clues to the difficulties which may be encountered when introducing artificially intelligent products. Both in terms of what the limits of their capabilities are currently and in theory, and in the reaction of people to using or working with systems which (who?!) transcend user friendliness to becoming a colleague. Gerhardt Freidrich from Digital Equipment Corporation's Intelligent Systems Group has been working on this issue. Freidrich believes that the introduction of AI systems in not analogous to the introduction of a new conventional software package. People have different feelings about working with and trusting the results from such systems. There is also the issue of threat we discussed earlier. He believes preparing people and organizations for AI systems will

---

33. Research on instinct does give some [applied] scientific support to Kant's views.

be critical for their mutual success.[34] This may become a serious issue, as there are already examples of people personifying AI systems, and the specter of the Luddites is with us still.

1.2.3 Mathematics

> The whole machinery of our intelligence, our general ideas and laws, fixed and external objects, principles, persons and gods, are so many symbolic algebraic expressions. They stand for experience; experience which we are incapable of retaining and surveying in its multitudinous immediacy. We should flounder helplessly, like the animals, did we not keep ourselves afloat and direct our course by these intellectual devices. Theory helps us bear our ignorance of fact.
>
> -George Santayana,
> The Sense of Beauty

> The Language of the Brain [is] Not the Language of Mathematics
>
> -John von Neumann,
> The Computer and the Brain[35]

> The laws we have to examine are the laws of one of our most important mental facilities. The mathematics we have to construct are the mathematics of the human intellect.[36]
>
> -George Boole,
> The Laws of Thought

Philosophers and computer scientists alike have looked to mathematics for a universally applicable metaphor to communicate across men and machines. Mathematics[37] provides languages and schemes for codifying rational algorithmic thought and logical relationships. Like so many intellectual

---

37. Including algebra, symbolic logic and other sub-disciplines. These will be dealt with in more detail subsequently.

matters, the ancient Greeks led the way. Aristotle codified syllogisms and Euclid codified geometry. There matters stood for more than a millennium. The rules of logic appeared to be a closed system, consistent—but unable to accommodate "non-logical" (i.e., associative) calculuses.

This does not mean that the "mathematics of thought" were unaddressed, at least in conceptual form. Gottfried Wilhelm Leibnitz (1646-1716) took Newton's accomplishment of mathematically describing the universe to the next conceptual step. Looking for a way to exchange ideas among men without regard to language, as Newton had made it possible to discuss the physical universe without regard to religion, he conceived the idea of a **Calculus Ratiocinator.** In this regard, Leibnitz furthered Hume's notion that the rules of thought could be codified. Leibnitz's concept of a calculus ratiocinator was prescient and from it stems directly Boolean algebra and such modern day tools as LISP, Prologue[38] and many of today's higher order AI tools.

Leibnitz's vision was given life by the English logicians Augustus De Morgan and, especially, George Boole (1815-1864). Boole's seminal work of 1854 **An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities**-usually shortened to **Laws of Thought,** developed a system of algebra, called Boolean Algebra. The great significance of Boole's work, for our purposes, was not only that it

---

38. Artificial Intelligence languages, covered in more depth later.

bridged the philosophical problem of Hume and Leibnitz, but that by its binary nature Boole's algebra was ideal for the digital circuitry of the next century. Although unknown to Boole, his system produced a way to mathematically represent information in the human nervous system, telecommunications networks, and the digital computer. As McCorduck writes:

> Symbolic and traditional logic are concerned with general principles of reasoning, but where traditional logic uses words, symbolic logic uses ideographs, which minimize the ambiguity of natural language.[39]

Boole's work was very influential[40] and was refined by others and reached its final form in Whitehead and Russell's **Principia Mathematica.** Lewis and Langford[41] view Boole and subsequently Whitehead and Russell's works as "...a landmark in the history of mathematics and philosophy, if not of human thought in general".[42] But as we shall see, all was not so simple. And well that this is so, for if it wasn't, it is unlikely that AI would have much of a firmament to be built on.

As is so often the case, new developments and discoveries for old dilemmas come in tides.[43] Classical mathematics was beginning to shake off

---

40. It even caught the fancy (No pun intended. Well...) of Lewis Carroll (a logician by trade) who concocted many stories and games which could be solved with such mechanized reasoning

43. I have drawn, as did Hofstadter, on Delong[44] and Nagel and Newman.[45]

its slumber. As theories of axiomatic sets and non-linear geometries developed, the Greek notions that there was only one kind logic and one kind of universe began to crumble. Work was done by the early nineteenth century mathematicians Carl Gauss, Wolfgang and John Balyai, N. I. Lobachevsky and, particularly, G. Riemann in describing and proving non-Euclidian geometries. Georg Cantor developed the **Theory of Sets** in the 1880's. Two Italian mathematicians, Gottlob Frege in Jena and Giuseppe Peano in Turin, worked to reconcile formal and associative reasoning by combining set and number theory. In Germany, David Hilbert worked on tightening Euclid's geometry to withstand more rigorous proof.

> I have hardly ever known a mathematician who was capable of reasoning,
>
> -Plato,
> The Republic, bk. I, 531-E

> The most beautiful thing we can experience is the mysterious. It is the source of all true art and science.
>
> -Albert Einstein,
> What I Believe

A certain culmination was reached, for our purposes, with Albert Einstein. The Special and General Theories established for [probably] all time that our universe really is associative and interdependent. Matter (physical) creates gravity (spatial). The physical creates the fields which affect the physical. Reality is a question of what you perceive. There are no fixed frames of reference. There is nothing which can be called "absolute", except the speed of light. There is a mysterious element to the universe.

Something deeply hidden had to be behind things.

-Albert Einstein

Just how is this related to Artificial Intelligence? Alan Turing, who we shall discuss shortly, maintained that perhaps whether a machine can think is a matter of whether one believes it can as a result of observation. The old empiricists would approve. Perception is reality.

> There are more things in heaven and earth,
> Horatio,
> Than are dreamt of in your philosophy.
>
> -Shakespeare,
> Hamlet, I, v, 166

However, even if all things could be dreamt of, it doesn't appear that they could by known by proof. That elusive philosophical line! As the nineteenth century drew to a close, the world of mathematicians, logicians and physicists had been greatly expanded. It was clear that there were many consistent axiomatic systems with which to view and understand both physical and intellectual worlds. The path seemed open to finding a system to codify thought. But the Greeks came back to haunt us again.

The Epimenides paradox came down from ancient Crete where Epimenides made the immortal statement: "All Cretans are liars." Derivative statements include:

- This statement is false.

- I am lying.

- The following sentence is false. The preceding sentence is true.

And so on.

These are examples of self-referential statements. The axiomatic[46] systems mentioned heretofore assumed that they could be proven **within their own systems.** Russell and Whiteheads' **Principia**[47] was a mighty attempt to derive all of mathematics from logic. That is to say, with total consistency from the system's own individual elements. But they were only able to do this by excluding cases of self-referential logic, or "strange loops", from their system. The Epimenides paradox not withstanding, the idea of such exclusions for the sake of [artificial] consistency troubled the mathematicians and logicians of the day and they spent decades trying to prove that **Principia's** system could be proven consistent **and** complete. If it would have been possible to do this, the Artificial Intelligence debate might have again be rendered moot because human thought obviously uses such self-referential logic.

David Hilbert cast the problem thus, and put it to the world of logicians and mathematicians for resolution:

---

46. Which is to say all mathematical and logical systems, symbolic or language based.

1. Was mathematics **complete,** in the technical sense that every statement (such as 'every integer is the sum of four squares') could either be proved or disproved.

2. Was mathematics **consistent,** in the sense that the statement '2+2=5' could be arrived at by a sequence of valid steps of proof.

3. Was mathematics **decidable.** Did there exist a method which could, in principle, be applied to any assertion, and which was guaranteed to produce a correct assertion as to whether the assertion was true.[48]

Hilbert. along with Russell, Whitehead, and most of their contemporaries believed the answer to these questions was yes. It is worth quoting Hilbert because his work represents the culmination of a line of thinking which began with Newton (See page 26.) and progressed through Leibnitz and many others.

> "In an effort to give an example of an unsolvable problem, the philosopher Comte once said that science would never succeed in ascertaining the secret of the chemical composition of the bodies of the universe. A few years later this problem was solved....The true reason, according to my thinking, why Comte could not find an unsolvable problem lies in the fact that there is no such thing as an unsolvable problem."[49]

Hilbert suggested that his questions be put to the test using Russell and Whiteheads' **Principia** as the subject. The basic argument for Artificial Intelligence appeared to have strong supporters.

The mathematical world was shocked by what happened only a year after Hilbert wrote the quoted words. The matter was put to rest by Kurt Godel (1906-), a Czechoslovakian mathematician. Godel's (paraphrased) theorem[50] is:

> All consistent axiomatic formulations of number theory include undecidable propositions.

Or—

> The statement of number theory does not have any proof in the system of **Principia Mathematica.**

So Godel proved there were unsolvable problems. There are no systems which can be proven by reference to themselves. So who is to say, or deny, that we can codify thought, or instill it in a computer. But Godel had left the door open on Hilbert's third statement. Perhaps there was a **mechanical** way to answer the question as to whether an assertion was provable or not, even if one couldn't actually construct such a proof.

The mathematical trail next leads us to Claude Shannon. Shannon, never an "AI" person, per se, nonetheless has become a key figure in its history. Shannon was a brilliant electrical engineer who did his masters work at M.I.T. and then became a legendary figure at Bell Labs. His thesis work and some of his subsequent work at Bell Labs were investigations into the nature of electrical relays and switching systems.[51] Shannon's novel use of Boolean algebra to describe the nature of circuits raised an interesting possibility. If Boole's "Laws of Thought" could describe electrical circuitry, could electrical circuitry describe thought? Many observers and historians of AI have highlighted this notion.[52] Shannon's work brought him into contact with Alan Turning.

---

52. I have drawn my information from Hodges[53] and McCorduck[54]

Alan Turing was an historic figure in many areas of mathematics, computers and "machine intelligence". Turing did an amazing amount of original work establishing many of the ideas, theorems and tests behind digital computers and Artificial Intelligence. It was Hilbert's third question (See page 40.) that sparked Turing's thoughts about problem solving or theorem proving machines. There is always a controversy about who "invented" the computer. There are really two issues; the concept, and the physical machine. The **concept** of what we now call the modern computer originated with Charles Babbage (1792-1871). Let us leave the discussion of physical machines for the section on automation. However, the precise mathematical definition, basis, and schemata of an **automatic electronic digital computer with internal program storage**[55] fell to Turing. His historic paper "Computable Numbers"[56] laid the mathematical definition of the concept of a 'machine'.

Turing's work in this area showed that there was no omnipotent machine, theoretically possible, capable of solving all problems. But more importantly, for our purposes, he showed that a machine could be conceived that could do anything a human computer[57] could do. He expanded this idea to a "Universal Turing Machine", which could do what any [specific purpose] Turing Machine could do. Turing was the first to develop a mathematically provable and consistent system for such machines. Turing

---

57. Meaning human's doing **computations.**

Machines and the Universal Turing Machines are analogous to special purpose and general purpose computers. With regard to computers, the idea of "Turing Machines"[58] has been with us ever since.

He is popularly known for breaking the code of the infamous "Enigma" cryptographic coding/decoding machine of Nazi Germany. His other raison d'entre is the "Turing Test". Even the casual reader of AI will encounter this test continually. It is often proposed, especially by AI advocates[59] as the definitive test of machine intelligence.

Towards the end of the forties controversy raged on the comparability of the human brain and the computer. A conference held in the philosophy department of the University of Manchester in England laid out the formal arguments, pro and con, on the question. The familiar issues of "mind/body", epistemology, etcetera were argued over again in the context of the reality of an emerging technology with the **potential** of being made intelligent.

Following this conference, Turing wrote a remarkable paper, "Computing Machinery and Intelligence,"[60] in 1950, advocating a very empirical view on the issue. This paper is also the source of the famous "Turing Test", though the notion had woven its way through his work since

---

58. A mathematical idealization of a computing automation. Used by mathematicians to define the idea of computability.

59. By AI advocates I mean those who believe that machines can be made to "think".

the thirties. The famous Turing test, roughly paraphrased, posits that if a person sitting at a terminal, by means of a dialogue between himself and another terminal, after a reasonable period of time, cannot determine if a machine or a human is composing and typing the replies, then the machine may be deemed intelligent. A machine which passed this test would, undoubtedly, not satisfy the solipsist. But it is the vexillum of the proponents of commercial Artificial Intelligence.

Computable Numbers led to Turing and Shannon getting together and discussing the idea of a thinking machine, with thought being captured in electronic circuitry. Having tackled the mathematical notion of computers (Turing machines) it follows that Turing was interested in what manner thought might be expressed therefore. In an amusing and incredibly ironic incident quoted by Hodges,[61] Turing speculates that a good test of such a machine would be if it could be taught to make buying decisions for stocks and commodities. This very idea is the topic of several companies developing expert systems today!

1.3 Theosophy

> The union of the mathematician with the poet, fervor with measure, passion with correctness, this surely is the ideal.
>
> -William James,
> Collected Essays and Reviews 1920

1.3.1 Cybernetics

Ever since Newton's remark in **Principia** (See page 26) people have searched for a philosophy and related calculus to understand and systematize human behavior and thought. This has occasionally led to some highly unusual constructs. Frequently they fail in their attempts at totality but end up making a valuable contribution to epistemology along the way. Analogies to electrical engineering have always been especially popular. The most significant attempt in this direction is **Cybernetic** and its father was Norbert Weiner. While Cybernetics is basically mathematical, it claims to be able to account for human behavior. Those who believe in it, have a bit of theosophy in their arguments.

Weiner's book **Cybernetics**[62]Cybernetics[63] built on work done in collaboration with Arturo Rosenbluth, a Mexican Physiologist, and Julian

---

63. Weiner defines this word to mean 'Control and Communications in the Animal and the Machine

Bigelow. Their paper, **Behavior, Purpose and Teleology**, published in 1943 cast the human nervous system as a machine built on the principle of feedback and servo-mechanisms; electromechanical devices which reached a high state of development during World War II. Cybernetics was an attempt to describe human activity in terms of well understood electrical phenomenon. Cybernetics proposed replacing the Newtonian notions of matter and energy with the notions of Information and Logic, as the basis of how things behaved. Weiner tried to transcend those who were looking for a calculus to describe thought in the Newtonian sense with a new system. Of course, people are not servomechanisms, and Weiner, et al, claimed to have found a way to embody purpose in their system. A particularly interesting aspect of Cybernetics was that it took Information Theory, developed primarily for telecommunications purposes, and generalized as a universal philosophy.

This thinking led to new attempts to construct logical systems and coding schemes to capture and then **program** thought. McCulloch and Pitts, a neurophysiologist and mathematician respectively, "A Logical Calculus of the Ideas Immanent in Nervous Activity" in the Bulletin of Mathematical Biophysics. McCulloch and Pitts drew from from the heritage that held that because the brain works between neurons on an electrochemical basis on "on/off"[64] basis, it should be possible to model it according to laws of

64. "All or none", —Boolean algebra and Claude Shannon's work again, though Shannon was never too taken with Cybernetics.

logic. Weiner's book, the work of McCulloch, Pitts and others in their path was very controversial in its time, and harkened up traditional fears about computers dictating to mankind, or worse, acting like him. However, like most proposed all encompassing systems, Cybernetics needed more and more modification in order to account for all the challenges which were directed at it. Cybernetics is better thought of as a movement than a science. Virulent and compelling in its newness and unorthodoxy, but ultimately hollow. The work of Weiner, Mcculloch and Pitts was a dead end. But much was learned along the way, and many were encouraged to enter the search.

1.3.2 The Church-Turing Thesis

Turing believed that thought processes were encodable. Harking back to Godel, while there might be unsolvable problems, apparently Turing, (and his spiritual descendants, did not think encoding thought was an example. Turing believed that computers could be made to "think" if they were "taught" how humans thought. Turing wrote;

> I believe that in about fifty years' time it will be possible to programme computers, with a storage capacity of about 10^9, to make them play the imitation game[65] so well that an average interrogator will not have more than 70 percent chance of making the right identification after five minutes of questioning. The original question, 'Can machines think?' I believe to be too meaningless to deserve discussion. Nevertheless I believe that at the end of the century the use of the words and the general educated opinion will have altered so

---

65. The Turning Test

> much that one will be able to speak of machines thinking without expecting to be contradicted.[66] But what "thought" was brought many of Turings contemporaries, and certainly their heirs through to this day, back into philosophy.

Turing was twenty five years too pessimistic on the advance of computer hardware technology. Many of the expert systems in existence today could meet his five minute requirement. But he may be spot on with his estimate of the turn of the century for the issue to have been rendered moot for all save the philosophers. Because Turing was primarily known as a mathematician, I have left the **Church-Turing Thesis**[67] to this section, although its central tenent lies between philosophy and mathematics.

Church and Turing developed the hypothesis independently during the thirties, but it was been called the Church-Turing Thesis in honor of them both. Church is worth an additional word. Continuing the tradition of Leibnitz's **Calculus Ratiocinator** (See page 34.) and Boole, Church devised his **lambda calculus**, "...an elegant and powerful symbolism for mathematical processes of abstraction and generalization"[68]. This achievement was an important milestone on the road of the propositional and predicate calculuses heavily used in AI languages and constructs today.

Let us look into the Church-Turing thesis a bit. It is the

---

67. Alonzo Church, prominent 20th century logician.

69. Philosophical or religious thought claiming a mystical insight into the devine nature.

theosophy[69] of those AI advocates who maintain that machines **can** be made to think.

The thesis has many forms, Hofstadter[70] distills it in three related ways:

1. What is human-computable is machine computable.

2. What is machine computable is FlooP-computable.[71]

3. What is human-computable is FlooP computable (i.e., general or partial recursive).

Hofstadter lists several variants on the thesis. Two which have the greatest bearing on this discussion are:

> **Isomorphism Version:**
>
> Suppose there is a method which a sentient being follows in order to sort numbers into two classes. Suppose further that this method always yields an answer within a finite period of time, and that it always gives the same answer for a given number. **Then:** Some terminating FlooP program (i.e., general recursive function) exists which gives exactly the same answers as the sentient being's method does. **Moreover:** The mental process and the FlooP program are isomorphic in the sense that on some level there is a correspondence between the steps being carried out in both computer and brain.
>
> **AI Version:**
>
> Mental processes of any sort can be simulated by a computer program

---

71. Stands for **Free LooP.** Hofstadter's whimsical theoretical computer language which is fully recursive, with unbounded loops. This concept is very important to AI. For various technical reasons, with the exception of ALGOL and the AI languages such as LISP and PROLOGUE, recursion is not permitted in computer languages.

whose underlying language is of power equal to that of a FlooP program—that is, in which all partial recursive functions can be programmed.

This leads to the general dogma of believers in true machine intelligence:

Hofstadter's AI Thesis:

As the intelligence of machines evolves, its underlying mechanisms will gradually converge to the mechanisms underlying human intelligence.

Implicit in this line of thinking is that the brain can be analyzed down to its chemical and electric (neural) functions. This reductionist view holds that if we can understand how cells and such operate, whether in the stomach or in the brain, we can build up any functionality of the brain from a recursive process. This has the effect of equating electronic substrates with neural substrates, both, one assumes, obeying the laws of physics. Thus identical, if isomorphically arrived at, results should be possible. This is the heritage of Locke and the rational empiricists. (See page 28.)

Where should the heirs of Berkeley, Hume and Hegel take succor? Their modern day counterparts are Hubert Dreyfus, Mortimer Taube J.R Lucas and Michael Polyani. Such men probably speak for most of the population in denying the notion that man is merely the sum of his physical parts; however microscopically one may build them up. This leads us to:

Church-Turing Thesis, Soulists' Version:

Some kinds of things which a brain can do can be vaguely approximated on a computer but not most, and certainly not the interesting ones. But anyway, even if they all could, that would still leave the soul to explain, and there is no way that computers have any bearing on that.]

Turing would have had a bit of a problem with this version, he was an atheist. Complicated world! Most of the severest critics of AI, such as Dreyfus[72] have their fundamental, and usually irreconcilable, differences with AI over such religious, sensory, and mystical criteria. This point-counterpoint goes on ad infinitum. As machines become more powerful and AI (or AI-like, you prefer,) programs are refined the debate may enter the class of unresolvable issues like most religious, philosophical, and metaphysical arguments usually do.

1.4 The Evolution of the Apparatus

> To err is human but to really foul things up requires a computer.
>
> -Paul Ehrlich,
> The Farmers Almanac, 1978

> The computer is no better than its program.
>
> Elting Elmore Morison,
> Men, Machines and Modern Times

The intellectual paradigms and the computer components of applied Artificial Intelligence have been converging since the inception of the machine. Occasionally tangential since the early nineteenth century, they have become intertwined since the advent of the computer in the nineteen forties. A context for the role of automation in the progress of AI is useful. Through the end of the forties, Artificial Intelligence theory had been able to develop independently along mathematical and philosophical lines for hundreds of years. In many ways, the well ran dry during that time. Empirical experience was needed to know where to drill next. No matter how expert the designer, sooner or later you have to built the plane to see if it can fly.

Automation became the gating factor. Even as late as the late-thirties work such as Turing's (See page 41) was viewed as interesting, but of little practical application. It was about at the end of that decade, fueled by wartime demands, that physical technology came into some rough alignment with intellectual proposition. The relationship has now come full circle. It is the state of computer technology which is the gating factor in the development of Artificial Intelligence. Because the state of computer technology is so permanently bonded to Artificial Intelligence and is just now attaining a power sufficient to create practical expert systems, it is worth a brief overview of its history and potential.

1.4.1 Complements and Catalysts

> All things are filled full of signs, and it is a wise man who can learn about one thing from another.
>
> -Enneads,
> bk. II, treatise iii, sec 7

The role of computers in Artificial Intelligence is catalytic and complementary rather than causal. It is a frequent occurrence in human activity that a breakthrough, or even a stochastic event, in one area leads to progress in an [apparently] unrelated area. The role of genetics in inheritance has been known and studied since Gregor Mendel's work in the mid-nineteenth century. Watson and Crick analyzed and documented the structure of DNA in the fifties. But genetic engineering was not possible until the seventies because applied microbiological techniques were not refined enough to permit it. In a similar sense, the evolution of computers enabled Artificial Intelligence to transcend the realm of the theoretical and enter that of the applied. Why did this occur when it did? Clearly, the notions of automating human thought and intelligence originate in antiquity.

The intellectual work was sufficiently advanced by the late thirties to blueprint theory of how "thinking" machines might operate. So why has demonstrable AI only been with us since the sixties and why are expert and knowledge based systems only now attaining commercial feasibility?

Gould has developed a theory called **Punctuated Equilibrium**[73] to explain major changes in the morphology[74] of species.

This discontinuous property of nature underlies most of physics as well as biology. The theory of quantum mechanics[76] shows that at the atomic level, matter reacts to and emits energy in packets or quanta. Atoms are required to absorb or emit radiation at certain fixed energy levels. The universe operates in step functions.[77] In other words a discrete amount of input is required in order to change the status quo.

I think a similar phenomenon exists for intellectual pursuits, applied science and mechanics. The progress we may expect in AI is going to be a function of advances in computer hardware. The progress of the hardware will be a function of an accumulated series of small quantum advances in semiconductor, logic design, miniaturization and a myriad of other

---

74. 1.The branch of biology dealing with the form and structure of plants and animals. 6. The form or structure of anything.[75]

76. At the risk of gross simplification.

77. These step functions can be exceedingly small, giving the appearance of continuity. But even in geometry one needs points to construct a curve.

technologies. The hardware, per se, functions independently of AI and to that extent is only complementary. It is catalytic in that increased hardware performance will enable AI researchers to test and implement approaches unfeasible heretofore.

1.4.2 Automation

> Deus ex machina [A god from the machine.]
>
> -Menander,
> The Women Possessed With a Divinity

> It is questionable if all the mechanical inventions yet made have lightened the day's toil of any human being.
>
> John Stuart Mill

Burke[78] traces the beginnings of automata back to ancient Greece, where the machines of Hero used water and air pressure to drive machinery for repetitive tasks Most commentators date the beginning of true automation to the Renaissance. The elaborate waterworks of the fountains of Villa d'Este at Tivoli or Chateau Merveilleaux led to automatic music played through water organs![79] Composers of such stature as Mozart and Haydn wrote for these instruments. These "machines", driven through differential water pressure, were programmable through the insertion of pegs into a mechanism of rotating cams and cylinders.

The gating factor in the history of advances in automation was usually the precision obtainable in machined parts, and the reliability of components of all types. Chemistry, metallurgy, and engineering led to many improvements within an established technology. This is easy to see in the history of timekeeping devices. The improvement in mechanical watches and clocks from the middle ages until the present time is entirely attributable to these causes. But often it took a breakthrough to advance society's capabilities by an order of magnitude. In this case, electronic timekeeping by means of oscillating circuits and then crystals. Very shortly after another quantum leap was realized with atomic timekeeping based on the natural vibrations of atoms and the emissions of radioactive elements.

The history of the steam power is another good allegory for it brings us to Charles Babbage, a man slightly ahead of his time, but **at** the head of computing. Building upon the work of Savery, Thomas Newcomen built the first practical steam engine. As the demand for more powerful engines grew, mechanical and metallurgical advances had to be made to accommodate the requirements of higher pressure engines. Brass led to iron, Watt optimized the components for more efficient use, Wilkinson improved machining techniques and so on. Horological and steam power advances made it possible to conceive of ever more powerful, intricate and precise engines—of all kinds.

Babbage's first effort was the **Difference Engine.** This was a table top machine designed to calculate navigation tables. A larger more accurate machine was commissioned by the British government, but it was never successfully completed, sacrificed at the altar of the inadequate machining techniques of the time. Meanwhile Babbage had turned his attention and energies to the **Analytic Engine.** This was to be a general purpose problem solving engine. Remarkably, it featured all the components of modern computers. Stored programs (A la Jacquard looms' control cards. Control cards survived right up through the computers of the 1960s.), a central processing unit of a generalized nature, and output devices. In these respects, Babbage was ahead of even the early digital computers of the 1940s, which were designed solely to produce bombing tables for the military. Unfortunately for Babbage, the [mechanical] technology of his day wasn't up to the task and he expended the balance of his life and most of his fortune trying to grasp what was within his mental reach.

1.4.3 The Digital Computer

> Computers can figure out all kinds of problems, except the things in the world which just don't add up.
>
> -James Magary

As discussed earlier (See page 41) computers are a result of mathematical and physical (engineering) advances. Special purpose mechanical computers had a long history before Babbage. Prehistoric man throughout Europe and the Americas constructed huge stone monuments to ascertain the comings and goings of seasons and celestial events. A thousand years before Christ, Arabic and Chinese[80] mathematicians and astronomers had saved several thousands of tons of weight over their ancestors with the

---

80. Who, of course, already had the abacus.

invention of Astrolabs for calculating the positions of the stars in the heavens, and derivatively, the equinoxes, solstices, seasons, eclipses and so forth. Significantly, such devices had the ability to take variable input data and yield results according to local circumstances. These devices were followed, albeit centuries later, by inventions of Pascal and Leibnitz, both of which could do simple digital calculations.

The nineteenth century saw great advances in communications. Morse with the telegraph, Bell with the telephone and, as a result of work by scientists such as Ohm, Faraday, Deforest, Henry, Fleming, Langmuir, Armstrong, Maxwell and Edison, and many others, advances such as radio, electromagnetic devices[81] and subsequently semiconductors came into reality. All these developments wove together, and infused with the urgent demands for cryptographic analysis and other other military mathematically based needs during World War II, led to the culmination of Babbage's dream of the computer.

Turing's contribution has been documented on the theoretical side,(See page 41) but it was under his direction that such machines—true "Turing Machines"— were developed at Bletchley Park during the war to break the Germany ciphers. Their names, Colossus, Delilah, ACE, and so forth bespoke the wonder in which they were held in their time, and as the first examples of machines with the capabilities to extend man's own capabilities and

---

81. Including servo-motors which operated on the feedback principle. Norbert Weiner used this idea as the basis of **Cybernetics.**

reach. The rate of advancement since the first ENIAC[82] computer has been unprecedented in human history. ENIAC had over 18,000 tubes, weighed over 30 tons, could "remember" twenty numbers at one time and could do 5000 arithmetic operations. It cost tens of millions of dollars. By comparison, today's IBM PC has only small silicon chips, weighs under twenty pounds, can "remember" hundred of thousands of numbers, can do over seven thousand arithmetic operations per second, and costs around five thousand dollars. Events moved very fast from hereon most of the story is well known and doesn't bear repeating here. But it is of great and ironic interest that many of the same names so central to the history of artificial intelligence played pivotal roles in the development of the computer.

If Babbage can be considered the father of the physical computer, Lady Ada Lovelace might be considered the mother of programming and the first to consider whether machines could "think". (See page 21.) Since that time, the issue of Artificial Intelligence has been as much a software issue as a hardware one. Since the application of Boole's laws and Shannon's ideas to electronic circuitry, it is possible to say the issue is almost exclusively a software one and that the contributions of hardware advances are primarily in that they make implementing more complicated (especially recursive) software possible.

---

82. Electronic Numerical Integrator and Calculator

1.4.4 Future Developments

The rate of development of the capabilities of computer hardware seems to prove even the most optimistic forecasters pessimistic ex post facto. VLSI design, content addressable memories, parallel processors and scores of other innovations suggest that the state of Artificial Intelligence will look to theory and implementation as the gating factors.

1.5 Palinode

> How much **can** the brain know? There are perhaps 10^{11} neurons in the brain, the circuit elements and switches that are responsible in their electrical and chemical activity for the functioning of our minds. A typical brain neuron has perhaps a thousand little wires, called dendrites, which connect it with its fellows. If, as seems likely, every bit of information in the brain corresponds to one of these connections, the total number if things knowable by the brain is no more than 10^{14}, one hundred trillion. But this number is only one percent of the number of atoms in [a] speck of salt.
>
> So in this sense the universe is intractable, astonishingly immune to any human attempt at full knowledge. We cannot, on this level, understand a grain of salt, much less the universe.
>
> -Carl Sagan,
> Broca's Brain

This introductory chapter has covered a lot of physical and intellectual ground. The goal has been to try to show that what we call Artificial Intelligence is not a new notion. Rather, it is a phrase which encompasses a cross section of man's best intellectual enquiry throughout the ages. The direction has been to focus the issues germane to considering what Artificial Intelligence might be, what its limits are, and what we may reasonably expect to be able to do with it in a commercial context. To summarize:

- What is it possible to know.

- How is knowledge acquired.

- How are decisions made (judgement).

- Can human thought processes be codified.

- Can such a "code" be imbrued in a machine.

- Could the necessary (physical) machine be built.

I do not believe that these questions will ever be definitively answered. But neither must we know why we are here in order to live our lives. Or whether pi ever terminates in order to be able to use it.

Most people do not understand how the technical things in their lives work. To be sure, there is usually an understanding at a superficial level. People have the notion that aircraft fly because air moves over and under the wings[83]. They do not know of Bernoulli's principle or Newton's Third Law of Motion. It is generally understood that automobile engines work by "exploding" gasoline in their cylinders. The mechanics of combustion and the mechanisms of the drive train are a mystery. People believe that antibiotics "kill" bacteria. That they do so by destroying the bacteria's ability to keep its cell wall intact, and therefore cannot reproduce or digest food and subsequently die, is not common lay person's knowledge.

In the introduction to this epistle I mentioned that people are not intimidated or particularly impressed by advances in technology. Society has

---

83. Although a surprising number believe it is because the engines "pull" the plane through the air.

become inured to even quantum leaps in technological legerdemain. This applies to computers with the apparent exception of Artificial Intelligence. People understand that computers can do arithmetic operations extremely fast, though they don't know how. They accept as a matter of faith that this ability is extendable such that symbolic manipulation[84] is possible (For instance handling words in a word processor).

But people have a problem accepting the notion that computers can be made to think as humans do. As discussed at length, while this is as much as philosophical issue as a technical one; its importance will grow in the public's mind as expert systems and Artificially Intelligent computers begin showing up in their lives.

> And God said, Let us make man in our image, after our likeness.
>
> Male and female created he them.
>
> The Holy Bible, Old Testament I:5, I:10

So people resist the idea of "intelligent" computers, at least on spiritual grounds. But as to whether computers can be made to **appear** intelligent, I

---

84. Of course, this is taking a liberty. Few people understand the significance of symbolic manipulation. People do not differentiate between numbers and letters. The computers is usually viewed as electrical substitution for the mechanical wheels of the old calculator. In one sense this is true, but numbers are treated as symbols, not as digits, per se. Thus people's acceptance of the computer's text handling ability is an implicit thought that "amazing" devices can easily do what (appear) to us as trivial things; rather than an explicit understanding of symbolic processing.

side with Shakespeare and Turing[85]. Shakespeare may have anticipated Turing:

> What's in a name? That which we call a
> rose
> By any other name would smell as sweet.
>
> -Shakespeare,
> Romeo and Juliet, II, ii, 43

Quot homines, tot sententiae

85. Hard to go wrong with that combination!

Chapter 2

The Anthropology AI

To successfully market commercial Artificial Intelligence products, it is not necessary to answer the philosophical questions. What is needed is a working definition of Artificial Intelligence. To paraphrase the Turing test: If we can build systems such that users **believe** they are talking to an "expert" in an expert system, or they really can speak[86] [type] natural language when querying a database or applications program, then for our intents and purposes, such systems are intelligent. As long as users will accept the idea of **Artificial** Intelligence without having to deal with "deus ex machina", there is reason to believe the time is right for commercialization. What is important is that buyers must be convinced that such products offer some desirable utility not available or useful to them in conventional software architectures.

Questions and skepticisms concerning AI usually arise among the business community from three sources:

---

86. Computer speech recognition and communications abilities are a special subset of AI technology. We will discuss this in the applications chapter.

1. Only a superficial understanding of the nature of a computer and how it operates.

2. Having seldom considered the nature of their own thought processes.

3. Ignorance of what the term Artificial Intelligence really means and confusion between AI based architectures and traditional algorithmic serially processed architectures.

This chapter attempts to put these and related questions in context as a necessary condition to better understand the potential, applications and differences of current and proposed AI products. Computers and humans are as different as the silicon and carbon upon which each is respectively based. A dialogue on the issue between three great minds:

> The computer is a moron.
>
> -Peter Drucker

> In Paris they simply stared when I spoke to them in French; I never did succeed in making those idiots understand their own language.
>
> -Mark Twain

> What we have here is a failure to communicate.
>
> Strother Martin,
> Hud

> Language is the light of the mind.
>
> -John Stuart Mill

Computer scientists and programmers would agree with Mill, Twain and Drucker probably reflect the views of most of us who have ever been frustrated trying to make a computer do our bidding. Everyone would quote Martin at one time or another.

2.1 Minds over Matter

She has no mind, and it doesn't seem to matter]

-George Burns

How do we control a machine do change its actions according to our bidding. How do we communicate with inanimate objects, give them a purpose and make them do our bidding.

It is usual to think of a computer as a very powerful calculator. Most people interpret that to mean that it is the latter day equivalent of a mechanical adding machine with electricity somehow replacing all those little brass wheels. This is a fundamental misunderstanding at the root of much of the confusion about Artificial Intelligence. Of course people understand that the computer can handle text and so forth. However people don't really consider why computers should be able to do this. And there, as the Bard said, lies the rub. Letters and punctuation are not arithmetic. No adding machine or calculator can manipulate them at all.[87]

The reason that computers can manipulate text is that they are **symbolic** processors. Arithmetic processing capabilities are a **subset** of their nature, not their sine qua non. Therefore numbers, objects and letters are equivalent. They have meaning to us as **representations** of abstract concepts. A "1" stands for an individual object or integer only in that we have assigned it such a meaning. Likewise "one" means "1" only in that we have assigned that word to represent that integer. "O" or "N" or "E" have sounds we assign to them and, put together, are pronounced "one", with its concordant meaning, because we have constructed a system with that rubric. "1", "one" or "O" have no absolute or universal meaning save that which we assign. The computer has no cognition. Thus it draws no conclusions and makes no inferences as a function of its existence. "Cogito, ergo sum"[88] does not apply. It can only work with symbols, meaning is a human notion. But the ability to work **symbolically** rather than merely **mechanically** (as an adding machine) distinguishes it from the mechanical calculators people often consider it an evolution of. The computer may not be a sentient being by virtue of its existence, but it is its symbolic processing nature which let it act as our calculating engine and hold the promise of teaching it imitate its

---

87. Some printing desk calculators can form words such as "total", "sum", etc. But these are burned into their memory as icons, and no manipulation is possible.

88. I think, therefore I am. Aristotle; also "Je pense, donc je suis"; Descartes.

creator.

Such symbols can be manipulated according to any arbitrary tautological set of rules. They can even be intermingled, and different classes of symbols manipulated according to different sets of rules. It is this unique symbolic manipulation ability that underlies the computer's possibilities as a processor of thoughts as well as numbers. The tautological rigidity[89] is an idiocrasy of a non-sentient being. It is hard to work with non-numerical concepts without out assigning explicit meaning or introducing perceptual and experiential biases. Only people trained as logicians can deal with a computer on this level. What is needed to accomplish both types of manipulation is a language and system of logic which enables the circuitry to capture both kinds of symbols, and structure the relationships both within and between types.

Understanding how computers operate is essential to understanding these issues, where we are on the learning curve of building intelligent software and what are reasonable expectations for product capabilities and development times. Key to this understanding is the notion of programming languages. Such languages exist on many levels and, taken as a whole, define the interface between human and the symbolic manipulation capabilities of the silicon. How comprehensive and transparent this interface can be is a function of how hard the "language" is for the human to learn

89. Pardon the redundancy.

and how much translation is necessary for it to be rendered in computer understandable form within the constraints on response time imposed by the users.

What the computer systems builder or user sees, at any level, is the highest level of abstraction available in that system, at that level. A moment's reflection will show that this is as true for the system builder as it is for the end user. End users like to work in a comfortable idiom, such as menues, interactive graphics or natural (like) languages. Likewise, the system builder desires to work at the highest level of abstraction available to him which still permits him the necessary flexibility to do the job. Both share the natural tendency togo with the approach of least resistance The expert (in his topic) user may get frustrated with overly simple interfaces because there is usually an inverse relationship between user friendliness and efficiency. In other words it takes more dialogue to communicate complex concepts when you are using an interface designed for the lowest level of user. The systems programmer has the option of switching to lower level languages. Within an application program or specialty language, the user generally does not have this option.

2.1.1 A Game of "Telephone"

Consider what would be necessary for an Englishman to speak to an Aborigine in a group consisting of himself, a Frenchman, a German, and an African. The Englishman speaks only English, the Frenchman speaks English and French, the German speaks French and German, the African speaks German and Aborigine. At least four levels of translation would be required to effect communication between the Englishman and the Aborigine. This is analogous to an unsophisticated end user dealing with a computer. From English to algorithm, from algorithm to application (complete program with all I/O specified), from application to programming language, from programming language to machine language. Obviously a cumbersome process taking many resources and fraught with the possibilities of miscommunications.

> The chief virtue that language can have is clearness, and nothing detracts from it so much as the use of unfamiliar words.
>
> -Hippocrates

This creates an additional challenge for those who write computer programming language and applications programs. People have a substantial capacity to withstand ambiguity in language. This is so because we have an ability to understand context, idiom, implicit meaning and the multiple connotations of words. Computers have no such [inherent] capabilities. Conventional computer languages must be able to account for whatever is typed in. Associative content cannot be inferred from symbols. Ambiguities usually necessitate additional queries for clarification.[90]

---

90. This is one of the greatest challenges for those researches in AI working in natural language recognition.

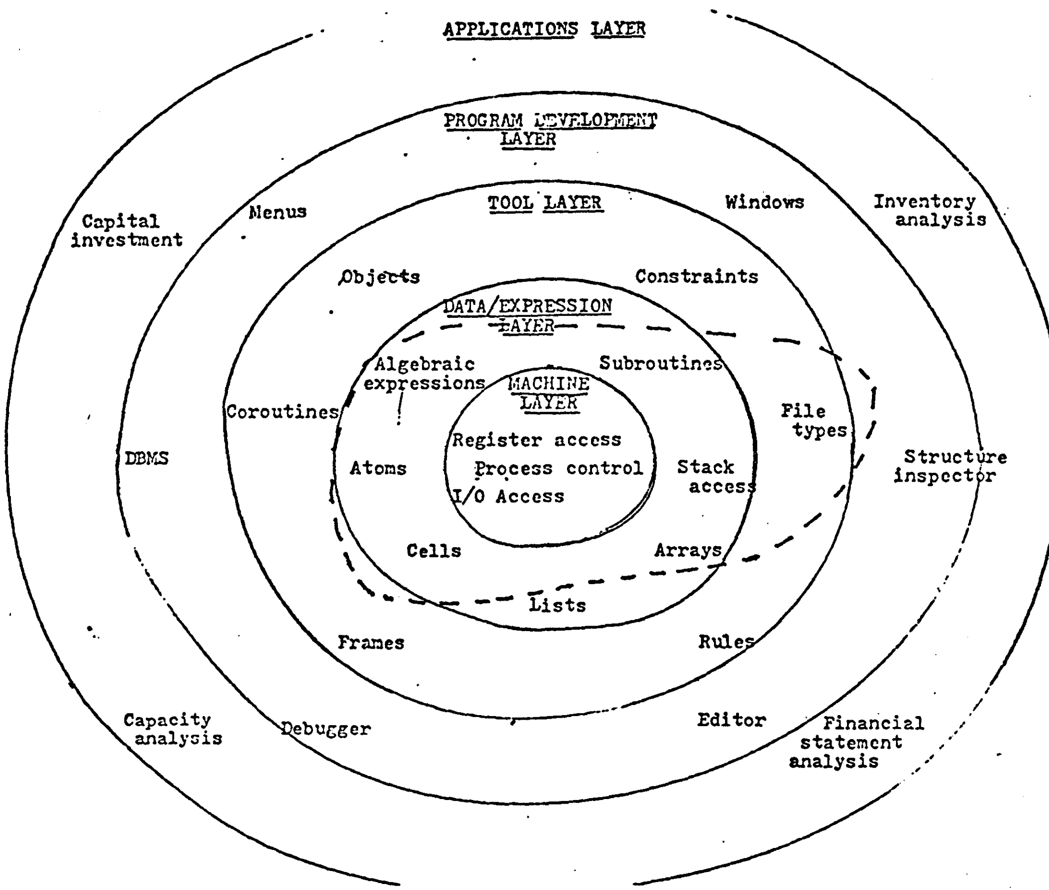
2.2 Digital Linguistics

2.2.1 Silicon Sign Language-The Machine Layer

Computers are constructed inside out. The kernel is the actual logical processing unit and all the associated and peripheral hardware. Surrounding the kernel is a very special language called **machine language** which straddles the frontier between hardware and software. Like layers of an onion, software is built up from machine language and at each layer, is transformed into a language more and more like human idiom. The entire point of over forty years of theory, research, and applied computer science has been to increase the cross-section of the onion. To bring computer language closer to human language, find ways to create single word abstractions, put more context and functionality into each layer and, ultimately, encode human thought processes into computer logic at an abstract level and thus enable humans to teach computers to mimic human own thought.

The AI Onion

Figure 1 is a representation of computer language used by Al Stevens.[91] This representation is drawn from the AI perspective. Lets work our way from the core to the skin in an attempt to gain an understanding of how and at what level we can expect computers to communicate with us, and visa versa. At the center of the diagram is the actual physical machinery of the


APPLICATIONS LAYER
PROGRAM DEVELOPMENT LAYER
TOOL LAYER
DATA/EXPRESSION LAYER
MACHINE LAYER
Capital investment
Menus
Windows
Inventory analysis
Objects
Constraints
Algebraic expressions
Subroutines
Coroutines
File types
DBMS
Register access
Process control
I/O Access
Atoms
Stack access
Structure inspector
Cells
Arrays
Lists
Frames
Rules
Capacity analysis
Debugger
Editor
Financial statement analysis

computer. Computers' symbolic processing abilities are rooted in hardware logic circuitry from which their central processing units are comprised. The basic man/machine interface is at the electrical level of these units. Here we require three things in order to enable these "chips" to do their work:[92]

1. A notation in which they can understand[93] our numbers and letters (symbols).

2. A logic which defines the meaning of the (symbolic) expressions.

3. A sequence of logical steps upon which to proceed through the expressions.

A computer can be though of as a collection of switches.[94] A switch can have two states, **on** and **off.** Thus a computer can have a theoretically infinite number of switches in various on/off positions. Therefore one could input, manipulate, store and output any kind of information which could be written in such a code. It should also be possible to define a logic for operating on such information in the same code.

---

92. This is not meant to be a complete representation of how computers are structured.

93. So to speak, "manipulate" is probably a better term.

94. In fact, this is exactly what a computer is like. Recall Claude Shannon's work mentioned in Chapter 1.

2.2.1.1 Mathematics

Humans calculate arithmetic on a base ten system. This means that we need the symbols zero through nine in order to express any number. But computers, as a collection of switches, can only manipulate two symbols, at their most fundamental level. Binary algebra is a mathematical system which needs only two symbols in order to express any number or symbol. Thus the sequence of binary integers is: 0, 1, 10, 11, 100, 101, 110, 111, 1,000.....etc.; represents the numbers zero through eight. Non-numerical symbols are just specially defined strings of zeros and ones. You can see that any number or symbol in one system can be exactly expressed in the other, although expressing a base ten number in binary could take many times more digits. CPUs have a series of binary operations built into them[95] which they can execute directly. Completely developed these instructions are known as **machine language.**

Binary numbers combined with statistical communications theory is integral to **Information Theory.** Developed by Shannon[96], Berger[97], Hamming[98] and others Information Theory is the blood in computers' and telecommunications' circulatory systems.

---

95. Such as read the contents of one register, add it to the contents of another register, and return the sum to the original register.

2.2.1.2 Logic For The Logical

> Contrariwise, continued Tweedledee, if it was so, it might be; and if it were so, it would be; but as it isn't, it ain't. That's logic.
>
> -Lewis Carroll, <u>Through The Looking Glass</u>

Having defined a basic idiom for the computer, a logic is needed. At the electrical level, computer hardware can only operate in a logical fashion. A switch can only be "on" or "off". Therefore a system of logic is needed, built into the hardware itself, which guides the computer in all its operations.

There are two logical systems required. One resides in the **arithmetic processing unit** within the larger CPU. This hardware is encoded, through a series of registers and process control, with the add, subtract, multiply, divide algebraic functions and their rules of operation.

Also necessary is a mathematical procedure for manipulating logical relationships in symbolic form. Boolean algebra serves this purpose. Although devised by Boole as a way to directly encode human thought, a purpose for which it failed, Boolean algebra is perfectly suited for encoding symbolic representations for computers. It failed for its original purpose because it was strictly concerned with formal logic and humans are not recursive. Boolean variables are confined to two forms—"yes" or "no". Obviously this is analogous to on/off or zero/one, the binary language of computers. The logic of Boolean algebra permits combinations of these on/off-or yes/no-choices to express the combinational and decision operations commonly needed to express equations, solve problems and write programming. Examples of these operators are:

- AND
- OR
- NOT
- EXCLUSIVE OR
- COINCIDENCE
- NAND (OR SHEFFER STROKE)
- NOR (OR PEIRCE)

Because all computer languages must be able to be compiled (reduced to machine language), Boolean algebra solves the problem of giving languages a logic appropriate to languages humans can work with as well as operations hardware can work with.

It is obviously cumbersome and very time consuming if one has to deal with a computer at the machine language level. We are analogue and frequently illogical creatures. The world of binary numbers and Boolean algebra is not our turf. Except in Information Theory binary digits (bits) are not data either. Binary arithmetic and Boolean algebra provide us an interface to the circuit level of a computer. What is needed next is a way to start to express "chunks" of data and actionable instructions in a single statement or short grouping of statements, yet still be able to directly translate into machine language in one step.

2.2.2 Operating Systems

The operating systems see to the scheduling of tasks through the processor, allocation and management of memory, and myriad tasks necessary to the systematic operation of the computer. One might think of operating systems as stage managers. Not actually part of the play, but critical to the staging of the play. Operating systems are programs usually written in assembly language. Different operating systems may be available for the same type of computer, each better suited for managing various categories of tasks. The next layers in the onion are the assembly and programming languages.

When a language, such as COBOL is said to "run" on a certain machine; what is actually meant is that COBOL has been modified to run on the specific operating system of that machine. The nature of these modifications is such that when a command with a known meaning in the language, such as "save", is encountered, COBOL can "call" the machine's operating system (transparently to the user) and arrange for that file to be saved. The actual saving and subsequent retrieval is accomplished by the operating system.

2.2.3 The Compiler

Compilers are special programs whose job is to translate a higher level language into machine language. Compilers translate assembly language or higher level language directly into machine language. In each case the computer's compiler breaks the complex language down into simpler and simpler components until it is directly translated into executable machine code. Compilers generally input high level languages and output machine code. Just as it is easier to to communicate if one can pronounce words and sentences as a flow rather than phonetically by syllable. Lets examine the special case of the assembler to understand how compilers do their tasks.

The assembler itself is a special compiler (translator) for translating the mneumonics of the assembly language into machine language. It is a special computer program which operates on symbolic input data to produce machine code instructions. An assembler generally translates inputed mneumonic codes directly into machine instructions item for item, and produces as output, the same number of instructions or constants which were defined in the input symbolic code.[99] The table of the assembler is in one to one correspondence with the actual instruction set of the CPU in the computer.

Assembly language allows the use of mneumonic names instead of binary values for the data and operating codes and also the locations of instructions and data. For example "D" would equal "divide", "M" would mean "multiply", "STO" would mean "store" etc. The question arises how the program "knows" that "STO" means "store", and further, how it determines that a certain string of binary numbers needs to be substituted for "STO" at the silicon level. This is accomplished because the assembler has a predefined table of these most primitive relationships built in. Just like hundreds of thousands of English words can be built up from only 26 letters, the assembler only needs a manageable number of elements in its table to translate the singular expressions into binary algebra.

Thus far we have moved from the kernel to the edge of the Data/Expression layer on the onion. In the early days of computers all programming had to be done directly in machine language. Assembly language was developed in the 1950s as a way to gain productivity. It should be obvious that higher levels of abstraction than machine language and assembly language became necessary to develop significant quantities of software.

2.3 Data Expression

2.3.1 Conventional Programming Languages

When you're lying awake with a dismal headache,
and repose is tabooed by anxiety,
I conceive you may use any language you choose to indulge in,
without impropriety.

-Sir William Schwenck Gilbert,
Iolanthe, act I

The Data/Expression layer transitions us from machine language elements to languages which permit more content per statement. One can extend this procedure of building the complex from elements of the simple to the idea of higher level languages such as FORTRAN, BASIC, COBOL, C, PASCAL LISP[100]. Such languages allow subroutines (e.g. generate a random number), algebraic expressions (e.g. {A * square route of B}), etcetera to be addressed (or "called") directly in the language. Each of the aforementioned languages incorporates functions and features [optimizing] it for its particular target audience. As you might imagine, there are many "dialects" of these languages as well. The development of conventional programming languages

---

100. FORTRAN-**FOR**mula **TRAN**slator, a language used for scientific programming; BASIC-**B**eginner's **A**ll purpose **S**symbolic **I**nstruction **C**ode; COBOL-**CO**mmon A**B**usiness **O**riented **L**anguage; LISP-**LIS**t **P**rocessor; PASCAL-A structured programming language named after its inventor originally conceived as a training language.

led to immense productivity improvements over machine and assembly language.

With such an improved capability over assembly language one might ask what utility machine language has at all. Sometimes the necessary performance can only be obtained by writing code at a more primitive (for the human) but more transparent (to the computer) level. For instance Basic programs often must have input/output routines written in assembly or machine language because Basic runs too slow to keep up with the I/O devices and the transmission speeds of the communications protocols. Compilers may not always translate higher level languages into machine code as efficiently as could be accomplished with assembly language. This is because they must take into account the idiosyncrasies of the higher level language and be prepared to accept the "worst" case. Where a highly repetitive or iterative operation is called for this can lead to greatly reduced response times. A clever (or desperate) programmer familiar with the design of the compiler can write these types of routines directly in assembly language to gain efficiency. Applications programs where response time is critical and many manipulations and transformations are called for are often written entirely in assembly language for these reasons.

The software industry has concentrated on developing higher and higher level (more "human" like) programming languages with which to write applications. The higher the level of the language, the easier it is to develop and perfect programs. For example, this is why COBOL[101] offered such a programmer productivity improvement over assembly language, and why the new "fourth generation" languages are supplanting COBOL. Each offered a five to tenfold productivity improvement over the next lower level language. There has also been robust development of special purpose languages such as **MUMPS**[102], a medically oriented language, **LOGO**, a learning language for children. Some languages are so specialized that they may not perceived as languages but applications packages. IFPS[103], Visicalc and many popular business oriented products are really specialty programming languages rather than applications programs even thought people seldom speak of "programming" in Visicalc rather than just "using" it.

The success of these languages made the dissemination of programming skills to a large, if still largely professional, section of the scientific and of business communities. Creating applications of all sorts became feasible in terms of time and cost. But the productivity improvements were for programmers. As demand the importance of information and demand for applications grew, data processing centers became power centers and programmers became a cult unto themselves. Almost by definition, programmer "types" had technical backgrounds and interests. As their own

---

101. For business oriented applications.

102. Massachusetts General Hospital Utility Multi-Programming System

103. Interactive Financial Planning System

importance increased programmers and their data processing managers often lost touch with the perspective of those they were writing their programs for. A gap developed between end users and the data processing people which became a matter of culture and language.

> The real danger is not that computers will begin to think like men, but that men will begin to think like computers.
>
> -Sydney J. Harriss

Companies wanted their computer programmers to use the primary languages to develop complete applications for functional end users.[104] Managers were becoming more frustrated and less tolerant of data processing departments and practitioners who guarded access to computers and were insensitive to end user demands. The demand for specialty end user languages also increased along functional lines.

> A tool is but the extension of a man's hand and a machine is but a complex tool; and he that invents a machine augments the power of man and the well being of mankind,
>
> -Henry Ward Beecher

> Give us the tools, and we will finish the job.
>
> -Winston Churchill,
> Radio Broadcast, [Feb. 9, 1941]

As computers became more powerful, smaller, and much less expensive they

---

104. By functional I mean staffs with specific responsibilities such as accounting, finance, human resources, etc.

started to proliferate throughout corporations. Mainframes led to minis led to micros led to the now ubiquitous Apples and PCs. As computers became accessible to functional departments throughout the firm, the demand arose for programming languages and applications packages designed for and usable by non-highly trained people.

2.3.2 Applications Packages

A dissatisfaction with the situation and an inability of even the most cooperative and sensitive data processing departments to meet demand led to the development of canned[105] **applications** packages and languages. Traditionally corporation created their own applications from scratch or hired consulting services to do so on a one time basis.. For many applications they were inventing—and maintaining—the wheel over and over again. In the early seventies vendors came into existence to develop, sell and maintain such common applications as accounting, human resources and project management software. It seems obvious to us now that this is a natural thing, but it wasn't at the time.

Companies specializing in such areas also leverage the accumulated **deterministic** expertise in the field. Such packages are designed to do a single function or group of functions "straight from the box", with the need for little or no modification or custom programming. Thus a company could

---

105. Meaning pre-written applications programs purchased from an outside vendor.

purchase a first class general ledger system from a vendor and know it was more comprehensive than that which he could reasonably expect to built. Furthermore, it was far less expensive because the vendor was amortizing his R&D and maintenance [106] expenses over an entire customer base. Different but related applications from a single vendor (such as general ledger and accounts payable) were designed to work with each other. Often one vendor will design his packages to be compatible with other vendors' products in complimentary or supplementary applications.

2.3.3 Specialty Languages

> Slang is a language that rolls up its sleeves, spits on its hands, and goes to work.
>
> -Carl Sandburg

Specialty languages are designed for particular tasks where the need is for a way to build applications in a specific area of functionality, such as finance, completely unique to a given company or situation. Of course this could be done in one of the conventional languages, but with a great deal more difficulty and requiring specialized programming talent.

Model building is a particularly good example. Spreadsheets are the most obvious. Everyone wants to use the same matrix construct. But in order to do that in conventional languages requires endless setting up of,

---

106. By maintenance I mean the never ending updating of the first release of the package to keep its functionality and usability competitive.

search of and reconfiguring of arrays. Additionally the matrix algebra to manipulate those arrays must be so specified. There are also many items of inconvenience, such as having to label cells in the matrix by algebraic rubric such as "A1" or "H7" in stead of "Revenue" or "Expense" The burden is on the programmer to keep house and know what the various arrays contain, pointer arrays to track intermediate results, etc.

The utilities are the same, such as the algebraic functions, various financial functions etc. But the arrangement of the elements is entirely unique. One company's model never is suitable for another pari passu. Contrast this with a general ledger package. The mechanics of double entry bookkeeping are well known, and also a matter of convention that companies must observe. The only real room for discretion is the naming of the chart of accounts. This is a trivial task and easily handled without the necessity for custom programming or formats unique to each company. Typical examples of packages and languages include:

Accounting	General Ledger, Accounts Receivable, Payroll, Auditing, etc.
Modeling Languages	IFPS[107], Visicalc, 1,2,3,, Simplan, Express.
Statistics	SAS, SPSS, Statman, etc. miscellaneous Human Resource management, Project scheduling, taxplanning, etc.

The key words for the 1980s has become **integration** and **end user**

---

107. Interactive Financial Planning System

computing. Instead of having incompatible programs, redundant data entry, etc; the notion has become to build applications up from a common structure. Instead of having to rely on the primary languages and their attendant staffs of specialists; the trend has been to develop even higher level languages and systems to allow the end users to create their own programming. Programmers haven't been neglected either. Sophisticated databases have greatly simplified storage, retrieval and program development. Fourth generation languages such as RAMIS, FOCUS, NATURAL and ADS-On LINE,[108] are built to work with the databases, have increased programmer productivity another order of magnitude. Finally, there has been an unmistakable trend towards purchasing software rather than building it. Especially where the purchased software is designed to provide a discrete functionality to a unit within the corporation. For instance, buying a proven general ledger package for the accounting department. The selection being based on accounting's analysis of the functionality and ease of use criteria.

2.4 Bounds of the Paradigm

> Every man takes the limits of his own field of vision for the limits of the world.

---

108. Products of Mathematica, Information Builders, Software AG and Cullinet respectively.

-Arthur Schopenhauer

All these developments have contributed to making the computer as ubiquitous today as it is. There are literally thousands of applications packages, hundreds of specialty languages and scores of database management systems and fourth generation languages.. The point is that people are becoming accustomed to dealing directly with (what they perceive) to be the computer. Every user of a spreadsheet is actually programming in a language especially designed to create spreadsheets. However this new diversity only partially ameliorated the explosion in the size and complexity of data processing departments. Great demands arose for support and extension of the applications packages. Invariably as the users of the specialty languages became more proficient, they started requests for interfaces between those languages and the company's databases and applications.

Different applications, even within one functional department, were often written in different languages, protocols, and for different computers and operating systems. This meant that one program could often not communicate with others even where common data was used, or where the output of one program was the input to another. The Tower of Babel incarnate. All sorts of patches and interface programs became needed. Redundancy sprang up, inefficiencies abounded, and the backlog now hovers between two and five years among the Fortune 500. These problems have become very time consuming tasks requiring cadres of skilled systems analysts, programmers, and layers of appropriate management. The demands have pressed against the limits of what can reasonably be done with traditional programming languages. Advances like databases and fourth generation languages are not the entire answer. They have slowed, but not halted the **rate** of growth, but not growth itself.

At this juncture it is worth pointing out the inherent limitations of traditional programming languages, their enhancements and heirs. The dashed line on Figure 1 delineates the capabilities of traditional programming languages. We will return to the limitations but for the moment we can understand that what the system builder can instruct these languages to do is limited to the languages' data structuring and procedural processing capabilities. These capabilities are bounded by that which can be explicitly stated with the standard rules of algebraic and matrix formats. However, compared to English, or even normal human syntax, these languages are unwieldy. While light years ahead of assembler, many many lines of code were still necessary to define even simple relationships and procedures. Symbols and relationships still had to be defined explicitly in mathematical form. Applications packages and specialty languages operate within these bounds. They just make reaching them easier.

The world of [conventional] computer languages today is rich and diverse. But we know that people's expectations continue to rise to meet and exceed technology. The call is for computers to transcend their serial 'mentality' and algorithmic approach to deterministic types of problems.[109]

We have seen that since long before the digital computer existed mankind has sought the cognitively intelligent automaton. The earliest pioneers of computer technology had this thought on their mind. The morphology of Artificial Intelligence has been evolving from the anthropology of the computer since Turing Machines were defined. An initial flurry of public opinion during computers' puerile days, fueled by Cybernetics, proved to be premature and naive. The disenchanted public turned away from Artificial Intelligence and it was cast in disrepute.

It fell out of mind because society became so enchanted by and addicted to what computers **could** do and there was such a long way to go to the horizon. But a generation of people and computers has now passed. People's expectations are coming around again to the idea of intelligent computers. They are looking for computers to evolve into robots, counsellors, tutors and even biological replacements in our own bodies. The generation so disillusioned with the visonaries of the late forties is exiting the stage. The generation born in the fifties got its perspective on computers during the space age. Computers have been amazing things, infiltrating our lives. They are now in everything from the Space Shuttle to the microwave oven.

---

109. The pedantic reader would point out that computers can easily handle problems with probabilistic elements. But the laws of probability are well defined and mathematically rigorous. Simulations will eventually stabilize about the mean. No heuristics get involved. The point is a technicality.

So it is natural that we look to machine intelligence as the next stop on what has seemed to us to be a smooth continuum of maturation over forty years that paralleled today's business leader's own lives. We look forward to Artificial Intelligence as the progenitor of the next generation of computer benefits to which we are entitled. It is time to molt the constraints of traditional programming languages and enter the brave new world. What is even more interesting is the attitude and assumptions of the children. Growing up with computers which are already highly developed and ubiquitous, children feel at home with them. The fourteen year old hacker breaking into the Pentagon's computer is already becoming a cliche. The next generation will expect to talk to computers as naturally as we expect them to give us our paycheck on time.

Chapter 3

The Morphology Of Artificial Intelligence

We have examined the history of Artificial Intelligence with regard to philosophical and mathematical issues. Following Turing, we have accepted an operational definition of Artificial Intelligence. Artifical Intelligence is just entering the commercial world. But much more research is needed to understand our own thought processes. We will examine the state of our AI epistemology, and where we are bound by the limits of our understanding of it.

This chapter examines the structure, languages and tools of Artificial Intelligence. The purpose is to differentiate AI languages and tools from conventional programming languages, and understand how AI products are constructed. It is useful to understand why some [potential] AI applications are within our grasp, and why others our beyond our reach. I would also like to set the framework for what technologies we can apply now, and where more research is needed before we can expect truly useful systems.

3.1 The First Tentative Steps

It takes two and a half to three years before human children are able to talk and communicate at a basic level. Education takes considerably longer, well into the teens at least. The reason why, of course, is because there is so much to learn. Not just facts and matters of rote, these are the easiest, but how to evaluate, analyze, gain experience, form perspectives, apply judgement and so forth. Learning is still a mysterious process, one we are just beginning how to understand. Needless to say, the task of making computers intelligent, teaching them both facts and reasoning power is a formidable task.

> The more people have studied different methods of bringing up children the more they have come to the conclusion that what good mothers and fathers instinctvely feel like doing for their babies is the best after all.
>
> -Benjamin Spock,
> The Commonsense Book of Baby and Child Care

Artificial Intelligence has evolved from an amorphous mass of theories and dreams to an indentifible entity with a life of its own. Interestingly, we can liken the growth and maturation of Artificial Intelligence to that of a human. After the first eighteen months or so of life, the human transitions from being a truly helpless and demanding infant, to an undoubtably minature version of its parents.

Speech starts to develop, pattern recognition is established, the first tentative steps are taken, and learning begins to take place.

> The desire of knowledge, like the thirst of riches, increases ever with the acquisition of it.
>
> -Laurence Sterne

Eagerly coached by the parents, the child is taught simple rules of conduct, how to frame request for food, and how to seek and give affection. As every parent knows, once intelligence becomes manifest and learning begins, even in the smallest ways, progress is rapid. The child's acquisition of knowledge is rapid, iteritive, and multi-dimensional. Parents are continually teaching and correcting the child's knowledge and behaviour (interface!) because of their desire to have the children respond as peers. Children are continually re-exposed to the same types of situations and tasks to deepen the child's understanding (reading or dressing, for example). Learning is multi-dimensional in that a problem such as solving a puzzle requires spatial, color, pattern recognition and associative knowledges and reasoning.

The progress of Artificial Intelligence can be likened to the human child's development. Putting the philosophical debates aside, research has concentrated on building up the same set of reasoning, pseudo-cognitive, and associative memory skills for computers as nature and parents do for children. As with children, a lot of time and effort is expended before results start to become useful. After thirty five years of research[110], most students of the field would place Artificial Intelligence's "age" at about eighteen months old; just completing the transition from infant to very small person. There are just a few exceptions to this, but as with children, once the signs of success begin to show, progress starts accelerating rapidly.

3.2 Meta-knowledge and Sensory-Cognitive Problems

The fact that Aritificial Intelligence has become something other than a laboratory curiosity today is due to the success researchers have had in duplicating some human inferential reasoning capabilities. In the last chapter, we discussed how such reasoning is accomplished throught the use of the Predicate Calculus, Semantic Networks, Frame, and Rule based systems; and in methods for dealing with uncertainty and "fuzzy" thinking. The areas which researchers have just begun to make progress in are what might be called the sensory-cognitive modes[111] of human intelligence, and meta-knowledge.

Meta-knowledge is knowledge an entity has about how it itself operates. As Barr and Feigenbaum describe it:

---

110. Albeit sometimes by just a few believers, with limited funds, in the "wilderness"

111. My term.

Meta level knowledge is simply the representation in the program of knowledge about the program itself—about how much it knows and how it reasons. This knowledge is represented in the same formalism as the domain knowledge, yielding a program containing **object level** representation that describe the internal world of the program, its self knowledge.[112]

For example, if we know how we have solved a problem, and our solution proves incorrect; we will concentrate on those areas where we know we were less confident during the original solution. Or when playing a sport, we know our own skill levels and best tactics, we therefore tailor our play to make maximum use of our strengths and minimize situations where our weaknesses are exposed. Meta-knowledge covers both AI and psychology.[113]

I use the term "sensory-cognitive" to mean the combination of input and evaluation which creates context. For example, we assign the meaning "hot" when our nervous system comes in contact with an object or emanations from an object (like a stove, or the heated air above a burner) which are greater than some threshold level. Beneath that level we might classify the sensation as ice-cold, cold, lukeware or warm. Above that level we might classify it as very hot, searing or scorching. Obviously the gradient is a continuum for a given domain. Observing a blast furnace, our use of the same word "hot" might have an entirely different calibration. We immediately understand the meaning of "hot" when we are cognizant of the domain. We can also communicate with others about the same domian and know that they understand the context and gradient appropriate.

This sort of thing applies to all the senses as well as to much imbedded meaning in the use of language. Psysiology and psychology still have much to learn about how humans sense and interpret such physical phenomenon. Whenever progress is made, it seems to highlight how much we don't know.

> Mankind, by the perverse depravity of their nature, esteem that which they have most desired as of no value the moment it is possessed, and torment themselves with fruitless wishes for that which is beyond their reach.
>
> -Francois de Salignac de la Mothe Fenelon, Telemaque, bk. VII

Hofstader quotes Larry Tesler, a researcher in the field, with this thought: "Once some function is programmed, people soon cease to consider it as an essential ingrediant of 'real thinking'. The ineluctable core of intelligence is always in that next thing which hasn't yet been programmed." Hofstadter has summarized it: "AI is what ever hasn't been done yet", and dubbed it **Tesler's Theorem.**

3.3 Applications in sight, but out of range

The following list of sensory-cognitive modes is adapted from Hofstadter.[114] The list includes breakdowns of specific areas where researchers have been concentrating.

1. Natural Language Understanding
    - answering questions in specific domains

- parsing complex sentences
- making paraphrases of longer pieces of text
- using knowledge of the real world in order to understand passages
- resolving ambiguous reference

Producing natural language.

- abstract poetry (e.g., haiku)
- random sentences, paragraphs, or longer pieces of text
- producing output from internal representation of knowledge

2. Speech Recognition

   - Understanding spoken words drawn from a limited vocabulary, (e.g., names of the ten digits)
   - Understanding continuous speech in fixed domains.
   - finding boundaries between phonemes
   - identifing phonemes[115]
   - finding boundaries between morphemes[116]
   - identifing morphemes
   - putting together whole words and sentances

3. Vision, Pattern and Context Understanding

   - recognition of individual hand printed characters drawn from a small class (e.g., numerals)
   - reading text in variable forms

---

115. Any of a small set of basic units of sound, different for each language, by which utterances are represented. (Random House Dictionary)

116. Any of the minimal grammatical units of a language that cannot be devided into smaller grammatical parts as, **the**, **write**, etc.

- reading passages in handwriting
- reading Chinese or Japanese printed characters
- locating prescribed objects in a photograph
- decomposition of a scene into specific objects
- recognition of objects portrayed in sketches by people
- recognition of human faces

4. Tactile— Touch, Smell, Temperature and "Mass" Understanding
   - identifing the nature of a surface (e.g., soft, hard, gritty, etc.)
   - identifing odors
   - understanding weight, mass

5. Creating original thought or works of art.
   - poetry writing
   - story writing
   - computer art
   - muscial composition
   - analogical thinking

Artifcial Intelligence projects which operate in the meta-knowledge and sensory-cognitive areas are at the forefront of research. But because of the inherent difficulties attendent out imperfect understanding of our own mental processes, and the primitive tools available for use to model them, progress is slow. Approaches to solve these types of problems are especially computationally intensive. To that end, the present generation of computer harware becomes a limiting factor. AI progress will depend on hardware progress. We will treat this more fully in the chapter on futures.

3.4 Knowledge Representation Today

> Intelligence...is the faculty of making artificial objects, especially tools to make tools.
>
> -Henri Bergson,
> L'Evolution Creatrice, ch. I

Computers are idiot savants. Fast, but without any understanding. Trying to teach them something out of their narrow, if incredibly proficient innate abilities, is a difficult job. The possibilities for AI applications are boundless, but as of yet, our reach exceeds our grasp. But progress has been made by concentrating on a more limited scope than parents and school teachers have to deal with. There are three major areas work has concentrated on and where sufficient progress has been made to built functional products:[117]

1. **Simple Knowledge Representation.** A collection of formula, experience, relationships and procedures.

2. **Task Resolution Methods.** Structuring of the problem and selection of the (best) currently feasible approach for resolution.

3. **Communications.** Human/computer interface. Input and Output

---

117. The following sections draw heavily from three sources; Kinnucan[118] Samuel Hotlzman,[119] and Hayes-Roth, et al.[120]

Workable knowledge representation systems must deal with three constraints.

3.4.1 Generality

The system must be able to deal with various types of knowledge. These include concepts, objects, formulae, procedural rules, etc. The following is a partial list of the range of knowledge types and thought processes humans use, which an [ideal] Knowledge Representation system must be able to account for, and which current systems must be robust enough so meaningful applications can be constructed:

Logic	By this we mean formal logic. Boolean and otherwise.
Heuristics	The "rules of thumb" we use to evaluate problems and situations, choose analytic methods, and interpret results. Heuristics include experience and judgement.
Knowledge Base	The sum total collection of heuristics, procedures, experiences and objects concerning the domain in question. The total knowledge base far exceeds any computer's memory. But more than just sheer storage, the amazing thing about the human brain is its boggling capacity to categorize and recall associations between objects, relationships, and experience. The computer analogue of the brain is the database

management system. While these systems are powerful, they are a fraction of the capacity of the human brain. Sagan[121] estimates that the brain has approximately 10^{11} neurons, each with one thousand associated dendrites. If each connection implies one "bit" of memory, it seems likely that we can "know" about 10^{14} "things". This is about one hundred trillion pieces of knowledge. The biggest computers today, with their associated peripheral memories are at least an order of magnitude smaller.

Further, our brains automatically "program" themselves upon incoming sensory information and conclusions of thoughts. Literally every item in a database management system has to be discretely specified for storage and retrieval. This is not strictly true, as relational systems can construct new lists from the merging and joining of existing ones without having each item specified. But the nature of the merges and joins must be specified. Such systems run exceedingly slowly when processing large databases. The best we can presently expect of an Artificial Intelligence system is to capture a bounded, domain specific knowledge base with enough memory to have utility for fairly discrete problem sets,

Learning

The system should be able to learn both from its experiences, and by direct extraction (or input) of new knowledge, procedures and relationships from the user, without expecting the user to understand programming or how the intelligent system itself operates.

3.4.2 Efficiency

There are basically three types of problems. Those which have a unique correct solution which can be solved algorithmically, those which have a solution which **appears** to be the best and emerges through repetitive simulation or heuristic search, and those which have many solutions which can only be heuristically arrived at.

Unfortunately, many problems that theory tells us have with algorithmic, deterministic optimal solutions may only be solvable by exhaustive search. That is to say they are only solvable for the best solution by calculating each possible outcome and comparing it explicitly or implicitly with the previous best solution to see if it is better. As we saw in chapter one, the set of possible solutions can become explosively exponential. Exhaustive search is not possible in a realistic period of time. These types of problems are much more prevalent than most people are aware of. Examples include:

- Most efficient routing, of trucks, transportation networks, process flows, etc.

- Many types of games, such as Chess or most card games.

The same can be true of problems best approached with simulation. Simulations require many iterations to stabilize, and many of those with multiple ques and/or complex mathematical programming formula can be extremely time consuming, if not combinationally explosive.

Then there is the class of problems which have **no** procedural solution, but are partly or totally heuristic in nature. For example, what color combinations "look" best in a given setting; what is the "right" amount of risk to take in a variety of situations, etc.

Efficiency is concerned with creating workable programs with acceptable response time. The best chess playing program conceivable is of no use to us if it takes 10^{95}th years to make its first move.

3.4.3 Unambiguity

The system must be able to understand us and we it. To do so it must have an unambiguous way of encoding our heuristic knowledge and non-algorithmic thought processes. It must also be able to sense ambiguities on **our** part, and itself, and through the solicitation of additional information from the user, be able to resolve them.

3.5 Paradigms of AI-Logic for the Illogical

> The central goals of Artificial Intelligence are to make computers more useful and to understand the principles which make intelligence possible.
>
> -Patrick Winston[122]

> Artificial Intelligence is the area of computer science concerned with the development of operational models of cognitive processes.
>
> -Samuel Holtzman[123]

Artificial Intelligence is a rich and diverse area of computer science research. The really isn't a "field" called Artificial Intelligence much more than there is a field called "computers". Researchers in AI field have been united by the notion of creating computers which can more closely emulate humans. Their goal has been to make computers more useful by making them more understandable to us, and in a mystical way, us more understandable to them. AI spans many different kinds of applications. It has enveloped various techniques and taxonomies for classifying knowledge, heuristics, and procedures. They fall into four general categories:

1. Natural Language Processing

2. Production Rules

3. Semantic Networks

4. Frames

5. Logical Constructions

We have seen how binary arithmetic and Boolean algebra define the lowest level interface between man and machine. But formulating the various combinations and permutations of ways which man expresses his thoughts and logic in Boolean algebra directly would be cumbersome. To address this problem, the concept of a logical system for humans (Leibnitz's Calculus Ratiocinator) has been developed and is called **The Propositional Calculus.**[124] An extension of the propositional calculus enabling it to deal with quantifiers is called **The Predicate Calculus.**

3.5.1 The Predicate Calculus

provides us with a formal system to represent logical propositions and relationships between propositions, suitable for deduction purposes.

In these logical systems, terms stand for the names of things and predicates stand for the relationship between things. For instance, think of objects like house, chairs, sink, bed and so forth. These are called **terms.** Examples of predicate names would be: is-a, inside-of, in. Thoughts, ideas and relationships can be constructed out of the predicate calculus. An example of how an English sentence would be stated in the predicate calculus:

Dukakis is Governor of Massachusetts

This is equivalent to:

Governor(Dukakis, Massachusetts)

Provided that we assign the meaning:

X is Governor of **Y**

To the formula:

Governor **(X,Y).**[125]

The predicate calculus has a highly developed set of laws enabling one to construct all possible combinations and inferences from statements made in the calculus. This is the logical basis for Artificial Intelligence.

3.5.2 Production Rules

Most people are intuitively familiar with the idea of a rule. In the form we all recognize it looks like this:

If it is raining out, take an umbrella

Or, more mathematically, as expressed in a simple BASIC program:

If A <= B then goto 200

Rules are generally in two parts where the antecedent represents some pattern and a consequent that specifies an action to be taken when the data matches the pattern. A typical rule in an AI system advising a manager on financing might be: IF cash flow is good, AND the company is paying no taxes THEN lease rather then buy to lower interest paid. When the rule is met, the system concludes that leasing will cost the company less than buying and adds that conclusion to its knowledge base.

3.5.3 Semantic Networks

Semantic networks are a method for representing relationships between objects in a knowledge base. They are somewhat analogous to database structures. In a network database architecture, numbers and words are linked to one another through embedded pointers. Each particular number or word could be considered an individual element, connected together they form molecules which have certain properties. Molecules can be linked together to form objects. The limitation of database technology is that the application program can draw no deductions or inferences from the structure of the data unless the programmer **specifies** how to manipulate the data for a desired result.

In an AI semantic network, individual elements are called atoms and are represented in the system by a string of alpha numeric characters. The character string "mortgage" would stand for the atomic concept mortgage.

---

126. Recall the Predicate calculus.

IS-A[126] stands for the mortgage's relationship to a class, and "debts" for the atomic class of debts. Complex objects are represented in memory by a connected list of atoms. Thus the fact that all mortgages are debts would be represented by by the list MORTGAGE IS-A DEBT. Recall that computers are symbolic processors. Hence they can manipulated these symbols without having cognition of what they mean. The predicate calculus provides the rules of manipulation.

Objects like loans, mortgages and debts can be linked together to form a network. For example: MORTGAGE IS-A DEBT, LOAN IS-A DEBT, DEBT IS-A LIABILITY. Such relationships may be envision as a network where the nodes represent the objects and the links represent the relationships. The network as a whole represents a taxonomy. Some versions of the semantic network incorporate inheritance properties. Thus LOAN inherits the property of LIABILITY through DEBT.

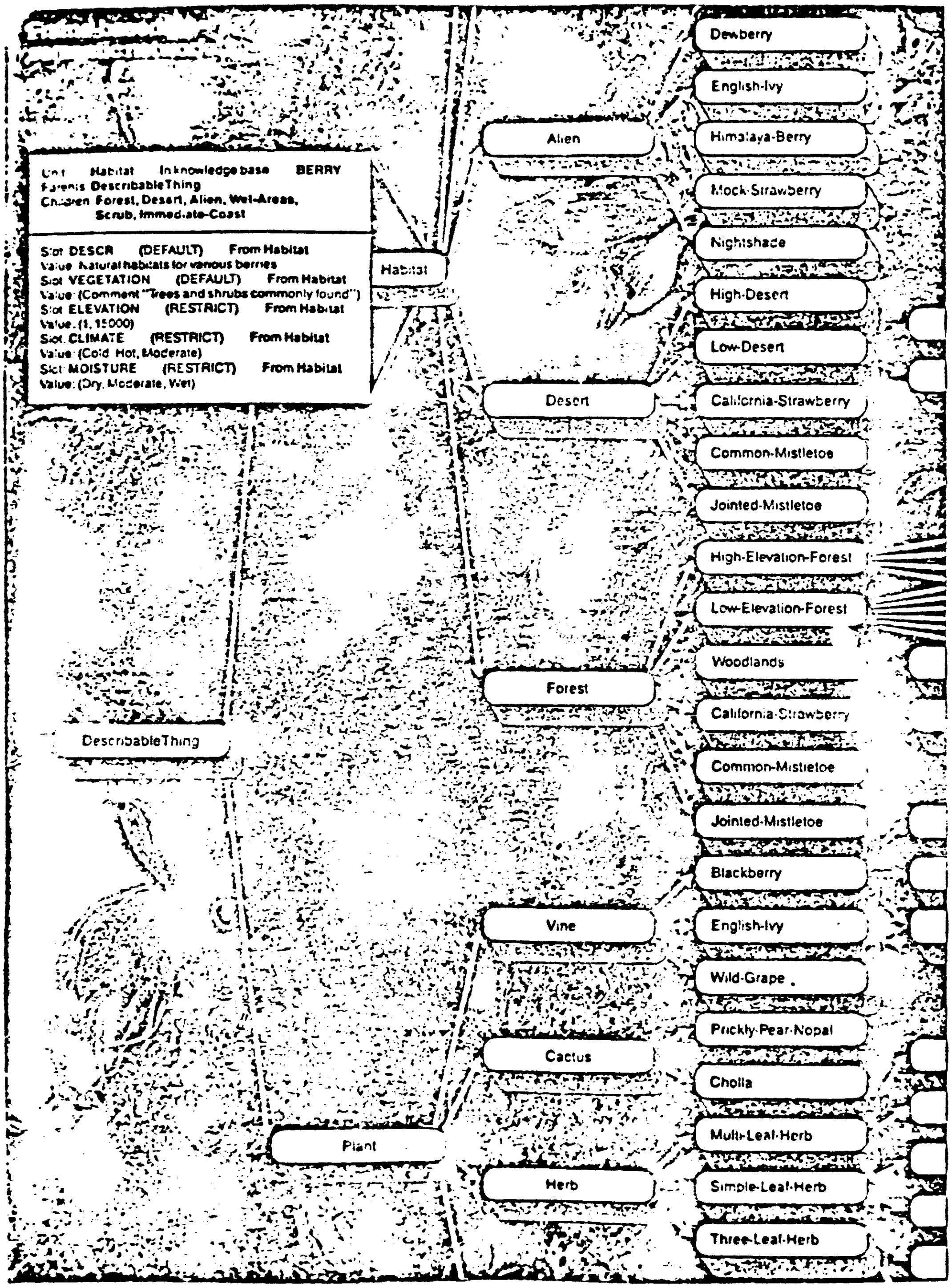
3.5.4 Frames-Object Representation

Hofstadter defines a frame as **A computational instantiation of a context.**[127] Frames are an artificial construct designed to allow the computer to capture the "gestalt" of an object by codifying the object's attributes and relationships to other objects. Returning to our financial example, a liabilities frame might represent mortgages as belonging to the class of LOANS and possessing properties such as INTEREST RATES, TERMS, DOWN PAYMENTS, etc. A frame system is a semantic network in which objects are represented by frames instead of atomic symbols. Thus each "node" contains knowledge about a complex object rather than an atomic one.

Significantly, the knowledge in the attribute "slots" can be numerical, and also "plugged" according to any arbitrary criteria. This is important because without this property, it would be hard for the inference engine to reach certain conclusions. This is easier to understand by returning for a moment to the rule based system. In a simple operation, one can deduce one's next step by comparing a logical operator either in algebraic or predicate logics. Thus if we encounter "A" we can compare it in some logical fashion to "B", determine the relationship (< > =, etc.) and proceed according to the predicate.

It is also necessary to have this property in a frame. Frequently that type of value is defaulted, but needed by a rule for input. For example, if the frame is for instrumentation, the object "oil temperature" may have a defaulted "normal" range. Thus the inference engine, operating on an input, can determine if the oil temperature is normal and proceed accordingly.

An interesting property of frames is that objects in a frame can inherit properties from more abstract objects. For instance if one classified a note as a form of a loan, the object note would inherit all the attributes of the class loans in the liabilities frame. Frames are, essentially, inherited semantic networks. An example of a frame based semantic network on the following page.[128]


Unit Habitat In knowledge base BERRY
Parents DescribableThing
Children Forest, Desert, Alien, Wet-Areas, Scrub, Immediate-Coast
Slot DESCR (DEFAULT) From Habitat
Value Natural habitats for various berries
Slot VEGETATION (DEFAULT) From Habitat
Value (Comment "Trees and shrubs commonly found")
Slot ELEVATION (RESTRICT) From Habitat
Value (1, 15000)
Slot CLIMATE (RESTRICT) From Habitat
Value (Cold Hot, Moderate)
Slot MOISTURE (RESTRICT) From Habitat
Value (Dry, Moderate, Wet)
Habitat
DescribableThing
Plant
Alien
Desert
Forest
Vine
Cactus
Herb
Dewberry
English-Ivy
Himalaya-Berry
Mock-Strawberry
Nightshade
High-Desert
Low-Desert
California-Strawberry
Common-Mistletoe
Jointed-Mistletoe
High-Elevation-Forest
Low-Elevation-Forest
Woodlands
California-Strawberry
Common-Mistletoe
Jointed-Mistletoe
Blackberry
English-Ivy
Wild-Grape
Prickly-Pear-Nopal
Cholla
Multi-Leaf-Herb
Simple-Leaf-Herb
Three-Leaf-Herb

3.5.5 Dealing with Uncertainty

The predicate calculus requires us to deal with finite elements. However, much of human knowledge cannot be put in a purely deterministic basis. It is stored in our brains in a shadowy form. We use the lens of context and experiential definition to bring it into useful resolution. Our judgement has many probablistic aspects to it. We also make decisions based on partial information and non-monotonic inferences. We clearly think and reason, but we can't describe the calculus which covers all our processes. The predicate calculus is a partial solution, but it is tautologically rigid and can't deal with uncertainty. Computer scientists are forced to turn to mathematics for methods to approximate such thinking.

Exactly how we do such thinking is imperfectly understood. Research is being done to provide Artifical Intelligence with mathematics to deal with unreliable data and knowledge. There are several approaches worth highlighting to get a feeling for how some expert systems deal with these kinds of problems.

3.5.5.1 Approximate Implication

Developed by Davis[129] and Shortliffe[130], approximate implication was used to assign certainty factos to heuristic rules. For example:

If this thesis is comprehensive,**and**
it is handed in on time, **and**
all the typing is within the MIT specified margins,

Then it is probably [0.9] that it will be approved.

The number [0.9] in this rule suggests that the evidence is strong, but not certain for acceptance. Evidence supported of the hypothesis is collected seperately from that which is discouraging and the "truth" of the hypothesis at any time is the algebraic sum of the evidence. There are some critisims of this method, centering on using alternative methodologies for assigning the probabilities or calculating their expected value.s[131]

3.5.5.2 Fuzzy Logic

Fuzzy logic is another departure from classic logic. For example:

Fuzzy Proposition: X is a large number

Corresponding Fuzzy Set; [X {0, 10},.1]
[X {10, 1,000},.2]
[{X>1,000},.7]

The interpretation of the proposition "X is large" is that "X might be less than 10" with possibility 0.1, or between 10 and 1,000 with possibility 0.2, and so on. The fuzzy values are intended to characterize an imprecise denotation of the proposition.[133] Approximate Implication and Fuzzy logic are just two representative examples of ways the Predicate Calculus and,

---

131. Such as Bayes' Rule, and subjective methods of assigning the probabilities.[132]

subsequently, the Inference Engine can be made to deal with uncertainty.

3.6 The Inference Engine

The various knowledge representation schemes address the problem of capturing concepts and associative relationships, in symbolic form, in a computer. Recall that a computer is inherently a symbolic processor. The function of the inference engine is to operate on the rules, frames, semantic networks, etc. to draw inferences from the relationships combined with inputed data and criteria. Once again, the predicate calculus is the underlying mechanism for guiding actions. To constuct human like reasoning engines, additional factors have to be taken into account.

3.6.1 Reasoning Mechanisms-Recap

Reasoning mechanisms are the core of Artificial Intelligence software. Much research has been done to gain a better understanding of how we think. The reasoning of logicians and mathematicians is precise. Each conclusion follows from the previous conclusions.[134] While mathematical logic has been well understood and highly refined over the course of hundreds of years, it is only of small use in understand how **people** think. People

---

134. Recall some of the interesting implications of the from the Chapter One discussion on Godel's Theorem.

aren't logical in the formal sense. This is probably because formal logic is suited to an abstract world where everything in axiomatic. The day to day world we live in is anything but axiomatic.

Expert reasoning is not monotonic. If it were, making decisions would be a tiresome thing as we would have to review every possibility, sometimes tortuously so, in order to reach a conclusion. Our experience and "commonsense" knowledge has no place in such a system. Most of our day to day decision making is assumption based. We think in many different ways, our initial choice of logic based on our heuristic judgement of what the situation at hand calls for. Sometimes we make assumptions, look for information to support our hypothesis, and discard and start over if we reach a dead end. Other times, we start with a set of facts and logically follow them at the logician would.

Human reasoning requires assumption, concluding on partial information, and dealing with uncertainty. We also must deal with new information which is uncovered in our search, and may not have been an antecedent condition when we started. A focus of Artificial Intelligence is how to enable computers, which are inherently unable to reason in such manners, deal with the necessity of it in intelligent applications. To summarize some of the aspects of human thinking and reasoning we have yet to know enough about:

- **Dependencies and Justifications.** New beliefs can result from new knowledge discovered or derived. When we operate on a theory, we sustain it with a set of beliefs we hold to be true from our experience or inference. Doyle[135] calls this phenomenon "truth maintenance". He has done some interesting work dealing with how maintain and modify our beliefs.

- Of particular interest is how we deal with ambiguity, and alternative courses of action.

- **Subproblems.** We often intuitively break complex problems into subproblems, solve the element and reintegrate them back into the larger problem.

- **Constraints.** We apply our a priori world knowledge in an integrative fashion to reject paths which we "know" will be unproductive. For instance we know not to waste time investigating how long it would take to walk to Florida from Cambridge because it is clear prima facie that the concept in untenable.

- **Metaproblems.** Hayes-Roth and Stefik[136][137] have done interesting work on systems that plan their own planning process. The notion is that systems which understand their own structure (humans, for example) plan problem solving strategies optimal for those structures. This "knowledge about one's knowledge" concept occurs throughout AI.

- For instance, work in control strategies is being done to design expert systems with the capacity to determine which problem solving approach is best suited to the problem at hand. This is one of the oldest notions in AI, originally being addressed by Herbert Simon in 1960.[138] The difficulty of emulating human ability in this area can be realized in that the challenge is still functionally unmet.

- **Uncertainty and Inexactness.** Researcher have long struggled with constructing systems to formalize human propensity to evaluate risks and operate under uncertainty. Bayesian statistics, Utility theory, and fuzzy set theory all have their advocates and detractors are models for dealing with this uniquely human phenomenon. Suffice to say that no dominant theory has emerges, though applications of these and others have proven useful in very bounded problem domains.

A thorough exploration of human decision making theory and approaches for formalizing it is far beyond the scope of this paper. The preceding was detailed to give a flavor of the complexity of the problem, and serve as a foundation for a late discussion of the probable limitations for expert systems. Key to understanding what inference engines can do and how they work is understanding the types of reasoning they can apply. There are three basic types.

3.6.1.1 Forward Chaining

Forward chaining, or data driven, reasoning reasons from facts to a solution. A forward chaining system defines rules that are capable of determining the correct values at every point in the problem given some inputs. Forward chaining is particularly effective where the are a set of "correct" rules which tend to produce useful results. An illustration[139] of a task suited for forward chaining reasoning is simplifying mathematical expressions. Here it is possible to determine a set of rules which will transform one expression into another, and proceed through them.

3.6.1.2 Backward Chaining

Backward chaining systems operate by searching back through a set of rules and knowledge base to find rules which, if "fired", would yield the conclusion at hand. This is an especially powerful concept which has been highly perfected for diagnostic purposes. An example would be medical diagnosis where the symptoms are known and the disease is sought. The backward chaining system would explore hypothesis about the disease by searching for rules, which if true, would yield the symptoms. Following this process to its origin, the disease can be identified.

3.7 System Building Tools

We have reviewed the structural nature of computer hardware, and how [conventional] programming languages were built up from its logic circuitry. None of these languages are directly suitable for use in Artificial Intelligence applications. Additionally, the closer to human language these computer languages get, the more there is a chance for ambiguity to creep in. They cannot handle the structures and relationships typical of human reasoning, except at the most trivial level.[140] Attempts to do more "AI" like tasks would require an unacceptable amount of cpu and memory resource, and would have too slow a response time. But the biggest hindrance would be the huge amounts of code necessary to describe AI structures. One of the greatest strengths of AI technology is its ability to handle heuristic search and problems with non-algorithmic solutions. Programming these (and other) types of structures in conventional languages would take an unacceptably long time. In order to handle such constructs, coding Artificial Intelligence languages and systems generators requires a very different approach. This is because Artificial Intelligence structures are concerned with a different agenda than conventional programming problems.

AI constructs have some characteristics which are hard to execute in

---

140. An example of this would be the one dimensional "if-then" statement.

conventional software.

- They are highly recursive.
- They use non-numerical symbolisms (at the user level).
- They are frequently associative rather than algorithmic in nature.
- They need unique logical systems.
- They are very screen and I/O intensive.
- They must be robust.
- They must self document.

The result is that such constructs are very resource intensive. Thus they have been a long time emerging from the lab. It also explains why much of the AI work to date has been has been done in universities and/or under government contract. It has historically been impossible to construct an operating AI product in even a mildly constrained hardware environment. However the great advances in semi-conductor technology have led to quantum improvements in hardware performance and equally dramatic reductions in cost. The net effect, in the past five years, has been to more than nullify the demands of higher level resource intensive languages. In fact, it has allowed them to proliferate.

3.7.1 LISP and the LIKE

LISP was invented by John McCarthy in 1958 for use in Artificial Intelligence. Few people realize that after FORTRAN, LISP is the second oldest programming language in widespread use. It has survived where many others are now gone or forgotten because of the validity and utility of its design for Artificial Intelligence. McCarthy[141] summarized the key ideas behind LISP thus:

1. Computing with symbolic expressions rather than numbers; that is, bit patterns in a computer's memory and registers can stand for arbitrary symbols, not just those of arithmetic.

2. List processing, that is, representing data as linked-list structures in the machine and as multi-level lists on paper.

3. Control structure based on the composition of functions to form more complex functions.

4. Recursion as a way to describe processes and problems.

5. Representations of LISP programs internally as multi-level lists, that is, in the same form as all data are represented.

6. The function EVAL written in LISP itself, serves as an interpreter for LISP and as a formal definition of the language.

LISP was meant to be a practical programming language. It was in the sense that FORTRAN or COBOL are. That is, practical for people trained in it. Because the world of AI was, and to some degree still is, an arcane one, the skill was not widely disseminated. It is a powerful, but low level language in that it is not user friendly.

LISP now has many dialect, such as QLISP, INTERLISP, and MacLISP. There are other AI languages which have sprung up in LISP's wake and garnered enthusiasts for special purposes, examples are SAIL, FUZZY, PROLOGUE and POP-2.

3.7.2 Intermediate Tools

Beyond the fundamental AI programming languages tools and specialty languages began to appear. Most of these were themselves built in LISP and are somewhat analogous to the specialty purpose languages we saw spring up on the conventional programming side such as IFPS or SAS. A representative sample:

Emycin	A domain independent, backward chaining, production rule oriented system useful for diagnostic and consultive applications.
KAS	Knowledge Acquisition System. A program designed to aid in the construction of rule based systems. It can handle probabilistic rules, and semantic networks.
Expert	Primarily diagnostic and consultive applications. Has the ability to choose from different hypothesis, based on available facts, and reach the most likely conclusion.
OPS5	A rule based programming language.

ROSIE	A general purpose system. Capabilities include, English-like syntax; forward chaining, backward chaining, and change driven.
AGE	A tool for knowledge engineers designing expert system. Lets developers work in rules and frames.

The entire list would be very extensive. A limitation of the list would be the systems are designed for thoroughly trained experts in the field. Many of them only can work in specific facets of knowledge representation and inference, most lack complete features for program development, debugging, and interfacing to other languages and databases.

3.7.2.1 Integration

Backward chaining, rule based, systems are the most widely used. Forward chaining systems are useful for certain types of problems, but they will never reach a conclusion if the inputed data and rules do not lead directly to one. Backward chaining will always always lead to a conclusion, even if it is a negative one; meaning that there is no hypothesis in the system which will support the inputed conditions. The key is to be able to integrate the various types of reasoning and knowledge representation paradigms in to a system robust enough to address the requirements of most expert systems.

3.7.3 The Next Generation-System Generators

To accelerate the development of expert systems, what is needed now is the AI equivalent of conventional fourth generation programming languages. Intelligenetics Knowledge Engineering Environment (KEE) and Inference Corporation's Advanced Reasoning Tool (ART) are examples of advanced AI systems generators just coming on the market. These tools, and other like them under development, promise to be the breakthroughs needed to see the rapid commercialization of Artificial Intelligence. Attributes of these new "systems generators" include:

- Multiple knowledge representation schemas including production rules, frames, logical/object representations, and semantic networks. All integrated so that one application can interweave representations.

- Multiple integrated inference engine capabilities including forward and backward chaining, formal logic, first order logic, predicate calculus, and probabilistic and fuzzy set capabilities.

- Well developed user interface tools, such as screen management, graphics, and mechanical user interface devices such as light pens, and "mice".

- Support for modular system design through multiple knowledge bases.

- Self documentation and explanatory features.

- Interactive debugging features combined with [limited] meta-knowledge for self diagnostics and program development

Systems such as KEE and ART should see commercial availability and acceptance starting in late 1984. They still must develop non-LISP runtime capability, portablity, and standard terminal support. Further, these are "first generation" system generators. In the future we will expect to see the results of additional research into meta-knowledge, non-monotonic, and fuzzy/probabilistic/heuristic methods of reasoning incorporated.

3.8 Communications

By communications I mean efficient and cognitive communications between user and program. Efficient in terms of rapid man-machine communication of content; cognitive interms of the machine understanding the human's a priori knowledge and context. Computer programs are usually heavily text and numerically oriented in their input and output functions. Further, the text in a computer language, is not readily understandable unless extensive "user friendly" interfaced have been written. Such interfaces are always less than ideal as they have no capacity to understand context, both linguistic and with regard to the state of the program at the time of communication. That is, they cannot explain themselves, or document and explain the session they have had with the user.

Successful expert systems building tools will need highly developed interface utilities including graphics, natural language, and mechanical devices such as mice and light pens. More importantly, they will require the ability to explain their logic and methods to the user so the user is satisfied with the "advice" and can update and correct the systems rules and knowledge interactively.

Chapter 4

Artificial Intelligence Taxonomy

AI is ready to step out of the lab, even if in swadling clothes, and start to be incorporated into products for commercial use. After years of research, Artificial Intelligence is ready to make its debut in the commercial marketplace. AI has applicability in almost every imaginable computer application, and will enable computers to serve many tasks it could not otherwise address. To understand the scope of the technology, I have broken it down into three very general categories:

- Natural Language
- Robotics and Sensory Processing
- Expert Systems

We will be primarily considering natural languages and expert systems. These distinctions are somewhat arbitrary, and there is significant overlap between them. Many applications require the use of more than one. The overlap is particularly great between natural language and expert systems. Many of the important aspects of expert systems are anticipated in the next section on natural language. The goal is to set the context for consideration of commercialization possibilities in the next two chapters.

4.1 Natural Langauge[142]

Natural language recognition is special domain in Artificial Intelligence. Natural language has been the focus of research since the field's inception in the mid-fifties. There are great difficulties associated with this area, indeed, as there are for its cousin; speech recognition. These are due to the problems associated with language's ambiguity, dependence on syntax, context, and definition. Natural language is an end goal, as well as a system building tool. As an end, it is mostly directed at database and [conventional] applications program query and analysis. As system building tools they will become a key interface compenent for expert systems.

4.1.1 The Need

In 1963, Artificial Intelligence pioneer Joseph Weizenbaum wrote a program called **ELIZA** as an experiment in creating a dialogue between a computer and a human. ELIZA emulated a conversation between a Rogerian psychoanalyst (played by the computer) and a patient. Weizenbaum intended

142. The terms "natural language" and "English" are used interchangably. The same issues apply for "foreign" languages as well

ELIZA to be a [somewhat] playful exercise, to explore the idea of natural language. Rogerian Psychoanalysis was chosen because the non-directive nature of such therapy provided a loose enough framework for the creation of realisitic sounding dialogue while still providing "content" sufficient to approximate a genuine human interaction. He specifically disclaimed any pretensions of therapeutic intent, content, or benefit.

Weizenbaum was surprised and then disturbed by what happened. People who used ELIZA were mesmerized by it. A prominent computer scientist became so involved in the dialogue that he started to reveal intimate secrets, **even though he "knew" he was talking to a computer.** Then professionals from the world of psychiatry began to insist that there really was therapeutic benefit to the program. Some other AI researchers extended ELIZA and, cooperating with some psychoanalysts, started to publish papers on its utility to psychoanalysis as a substitute or complement to human dialogues. The situation got a bit out of hand, and Weizenbaum strayed away from the idea, disturbed by some of its implications.

All this is by way of saying that people are fascinated by, and attracted to, the concept of communicating with machines in English. We all have an innate desire to understand and be understood. Most of us are also find communicating with computers as frustrating as trying to communicate with a visitor from France; when all the French we know has been accumulated from restaurant menues and nursery rhymes.[143]

None of this has been lost on the purveyors of commercial software, especially decision support software aimed at non-computer literate executives. Reading the marketing literature of many computer software vendors, one could easily believe that it is possible to "talk" to computers as easily as Doctor Doolittle talked to the animals. "English language" capability has joined terms like "user friendly", and "interactive" on the scrap heap of words which no longer have any meaning. It seems that all kinds of computer products are as literate as Edwin Newman and as easy to talk to as your next door neighbor. Unfortunately, computer natural language progress is closer to Casey Stengel than Edwin Newman.

Observers have commented that the real reason for the explosion of the personal computer was not the invention of the microprocessor, but the writing of programs that had utility for, and were usable by, non-computer literate decision makers. While the point may be arguable, most would agree that few purchasers of personal computers didn't start off with a spreadsheet or word processing program. The ideas behind these programs weren't unique. Modeling languages and text editors had been around on mainframe computers for years, but they assumed some computer literacy. The notion is that the user interface was the key to their success. Shwartz has observed that "..much of the progress of software technology can be viewed

---

143. e.g., "Frere Jacques", or "Canard a L'Orange"

in terms of the increased user-friendliness of data processing systems."[144]

Natural language is important because it is a key part of human intelligence, not merely an interface to it. As Bates and Bobrow state it: "...it is difficult to separate linguistic capabilities from other human capabilities such as memory, reasoning, problem solving, hypothesis formulation, classification, planning, social awareness and learning."[145]. Natural language is hard to precisely define, Woods has neatly defined it, vis a vie computer languages: "Natural language assumes understanding on the listener's part, rather than mere decoding. It is a vehicle for conveying concepts such as change, location, time, causality, purpose, etc. in natural ways. It also assumes the system has an awareness of discourse rules, enabling details to be omitted that can be easily inferred."[146]

4.1.2 Expectations

Woods' view is a key to understanding why natural language is [ultimately] what is needed to realize the full potential of computers to act as "people amplifiers". To transcend today's deterministically based software[147], we must look to Artificial Intelligence. Our expectations[148] for natural language and expert systems include:

---

147. Such as databases, decision support systems, functionally specific applications software (e.g.' accounting packages, payroll, etc.). While there are various kinds of software which use stochastic processes and simulation, they are still algorithmically based; "judgment", or non-algorithmically based saolutions, can not be incorporated.

- Capability to understand the user's general and domain idiom, and converse in a natural, interactive fashion.

- Knowledge of the user's **expectations** of what the system can do.

- Knowledge of the user's intentions and goals.

- Knowledge of the domains appropriate to the problems at hand.

- Knowledge and Interpretation of the user's familiarity with the system. That is to say, diagnosing the level of the user's interface with the system, and adjusting the interaction according to assure the user is properly served.

These are very important criteria, and directly related to the assumptions people make about the nature of intelligent conversation with an expert in a field. To give some idea of the scope of the task ahead of researchers, I return to Woods[149] for detail:

- Request an action by the system or an effect to be accomplished where the level of the description in the request is abstract and details are filled in by the system.

- Ask questions whose proper interpretation depends explicitly or implicitly on the system's ability some of the user's intentions.

- Propose modifications of previous requests or of system responses where the system is to infer the relationship between the modification and the previous discourse.

- Ask for clarifications, and then modify a request, where the system provides the help in response to the request for clarification and properly responds to the modified request.

- Order the system to modify its overall future behavior, where the system responds by changing its internal model of future action to conform with the order. [Meta-knowledge]

The preceding lists might be considered sort of a "general intelligence" specification. Operationally, the system must be able to:

- Interact with the user in a natural way (per above).

- Characterize the problem.[150]

- Bound and quantify the problem.

- Locate the relevant information (Even to the extent of communicating with other expert or non-expert systems).

- Present the information in an understandable and useful form.

---

150. By this I mean, examining the problem in a top down fashion, and then detemining one or more approaches for solution; trying them out, and then selecting "the best" way to as the one to be used. Of course, it may be desirable to show the user this decision process where alternative methods yield significantly different results and then make a recommendation according to the system's "judgment"

- Learn from its interaction, and add that experience to it knowledge base.

4.1.3 Requirements for True Functionality

In order to accomplish these things, efficient informed communication is absolutely necessary. Without this, the user may have to be specific beyond his [going in] knowledge of the problem; or be expected to provide guidance to the system which he expected the system to provide to him.[151] Alternatively, extremely lengthy dialogues may be required to lead a [too] generalized system to the area the user wants to explore. Ideally, an expert system will have deep enough understanding of the user and the domain to be an intelligent advisor rather that pedantic or ignorant interrogator.

To begin to attain the expectations and required functionality, natural language developers must address very detailed issues of how we communicate and develop ways to emulate them in software; for example: 152

- **Comprehensiveness.** A measure of how close the system is to complete conversational English. Elements of comprehensiveness include:

  * Lexical Coverage. Size of vocabulary and appropriateness to the

151. The classic "blind leading the blind" situation

152. Adapted from Bates and Bobrow[153]; and Shwartz.

application domain.

* Syntactic Coverage. Proper handling of complex verbs, relative clauses, various question forms, comparatives, time and place relationships etc.

* Semantic Coverage. How much does the system **understand** about the domain. Does it have a model of the domain, or merely translate English questions into specific queries in a formal query language?

- **Habitability.** A measure of how quickly and comfortably a user can recognize and adapt to a system's limitation. Comprehensive systems may be able to answer most questions, but possibly not in a "natural" manner. Less comprehensive systems may actually out perform more comprehensive ones, if they are better designed for the domain and application in question.

- **Resiliency.** Handle unusually or ungrammatical requests. ("I ain't got none"); complex and overly complex requests ("Would you be so benificently forbearant as to procure the derivative results of the failure of our pubescent division to cleave its strategic plan?")

- **Anticipate the user.** Discern, interpret, respond to and take corrective actions, through additional clarification dialogue or knowledge search, when misconception and misunderstanding is detected in the interaction due to:[154]

* "Extensional" Misconceptions. These occur when the user assumes, implicitly or explicitly, that a subset of a known class exists; and takes a general answer as confirmation. For example, If the user asks "how many Sloan Fellows failed operations management" he is assuming that some Sloan Fellows **took** operations management. If the system answered "none", he might draw conclusions which are not valid.

* "Type Misconceptions". The user assumes a relationship exists, when it does not. For example, "How many Sloan Fellows **teach** operations research?" Assumes that Sloan Fellows can teach the subject.

* "Object Related" Misconceptions. Discrepancies between what a user believes about an object, and what the system believes. The user might posit that "Sloan Fellows always get perfect scores", the system should dispute that.

* "Event Related" Misconceptions. The system must be able to discern changes in status for a situation resulting from new information.

4.1.4 Constraints on Natural Language

Clearly, successfully implementing natural language on a computer is a very difficult task. It is the farthest from any commercial reality. Even a speaker fluent in English, is unlikely to be fluent in every dialect and special idiom. People from Great Britain are fond of saying that we in America speak something called "American". It can be interesting to observe someone from Ireland trying to communicate with someone from rural Mississippi. They can work things out, but a lot of the conversation will be devoted to defining terms and placing contexts. Each will find communication much more difficult compared with the ease, familiarity and transparency of talking with someone from their own "domain", even though each is speaking the "same" language.

When dealing with computers, there is usually an inverse relationship between size (flexibility and comprehensiveness) and response time. This is especially true when relationships are stated or stored in multi-dimensional fashions. This is why relational database systems operate slower than hierarchical or network architectures. The former require many more computations to obtain the data, the latter race down a series of pointers. This is roughly analogous to the difference between someone must find his way by reading a set of directions, and then referring to a map; and someone who follows a series of "arrow" signs specially set up to guide him to his destination. Today's computers are von Neumann machines, they can only operate in a sequential fashion. Thus tasks which are computationally intensive slow down as a function of the complexity. Computers cannot [yet] "walk and chew gum at the same time", so to speak. However, computer power has been growing at a fantastic rate, and we are on the verge of seeing parallel processing become feasible. Increased horsepower is clearly a component of creating faster and more powerful natural languages.

4.2 Other User Interfaces

While natural language interfaces are very important, they are not omnipotent. In some cases they are suboptimal in that it may take **longer** to arrive at the answer than through less "advanced" technological means. For example, there are many situations, such as using the cash machine at the bank, where a menu is more efficient than a natural language interface would be. In fact, more commercially oriented work in expert systems today is away from natural language and towards intermediate and highbred interfaces.

Natural language can be a great disadvantage where physical interaction and rapid processing of analogue information is required. It is hard to imagine a computer driving a car by having a passenger type in a description of the road ahead, or playing a video game even if speech recognition where possible at high rates. (A little to the left, no the right, no the left....crash!)

Graphics is frequently more useful then language, especially where the users understand the meaning of symbols and icons. Xerox has done years of research on the subject at PARC. This is the theory behind Apple's LISA and MacIntosh. Graphic information display is frequently coupled with physical interpretation and response. The use of graphics is well proven in flight control and navigation applications. All the situational information about the aircraft's spatial position and vectors is displayed on symbolic instruments as is the weather information on the radar and the systems status on a variety of analogue guages. Physical interaction are often better accomplished with eye hand coordination. That is why video games frequently have joy sticks, and the mouse is becoming so popular.

True natural language will probably never be implemented on a computer because language **is** part of intelligence, not just a way for a sentient being to communicate its intelligence. Natural language will play a key role in Artificial Intelligence applications. However, it is still quite a way from being ready to be generally used as an interface technology.Even a very robust AI natural language implementation may not always be desirable because it may be too inefficient for the task, or not able to handle the specificity of some domains. It is also not the most desirable interface for all purposes, especially for those which require extremely rapid man-machine interface, or where there information which is better communicated in graphic form.

Pure implementations of natural language will be most useful in very specialized applications. Natural language will find its greatest utility in combinations with other interface technology where the program designers can call it where it is most appropriate, and the users can use it when that this their preferred way to work.

4.3 Expert and Knowledged Based Systems

Knowledge is Power

-Francis Bacon,
Meditationes Sacrae

As long as humans have been on this earth, we have always sought to learn more than we know from our own experience. In Biblical times Pharaoghs surrounded themselves with wise men and magicians.[155] In Grecian times people would visit the Delphic oracle to seek knowledge, advice, and for guidance about the future. History is full of references to alchemists and soothsayers. The world was simpler then, most of the advice sought was on how to resolve personal problems, please the gods and so forth. Today almost every businessman and professional has a cadre of seers and viziers who will be happy to aid and advise, for a price. They are called experts and consultants.

True experts are much more valuable. An expert is someone who has a particular skill or knowledge in specialized field. There are few experts on "life", but many on strategic planning.[156] Because expertise, by definition, is a scarce commodity; demand always exceeds supply. The luxury

155. We know how much help they were when God got annoyed.

156. At least at MIT.

of expert assistance can only be justified when the stakes are high. This doesn't render it any less desirable, just unavailable for the non-specialist facing the task every day. The president of a small company would like to have a staff of financial experts to aid him in capital budgeting, the plant manager would like to have a staff of operations management experts to optimize his work flow. But while such expertise is available, it isn't feasible or cost justifiable to have these and other types of experts available at a moment's notice. Even if they were, it would take them time to "come up to speed" before they would be ready to address the problems at hand.

The world has gotten more complex. Every profession has scores of specialties. Our lives and jobs are more complicated than ever before in history. It is no longer possible for us to be a Francis Bacon, and set out to know everything that is worth knowing. The current interest in expert systems is a response to these needs. Conventional software can perform tasks, but it can't have knowledge and reason. What is an expert system? DARPA[157] is a government agency keenly interested in artificial intelligence and expert systems. They define an expert system as "...the codification of any process that people use to reason, plan, or make decisions as a set of computer rules."[158] DARPA's definition and those discussed previously in the natural languages section should give the reader a pretty good idea of what

---

157. Defense Advanced Research Projects Agency

an expert system should be able to do.

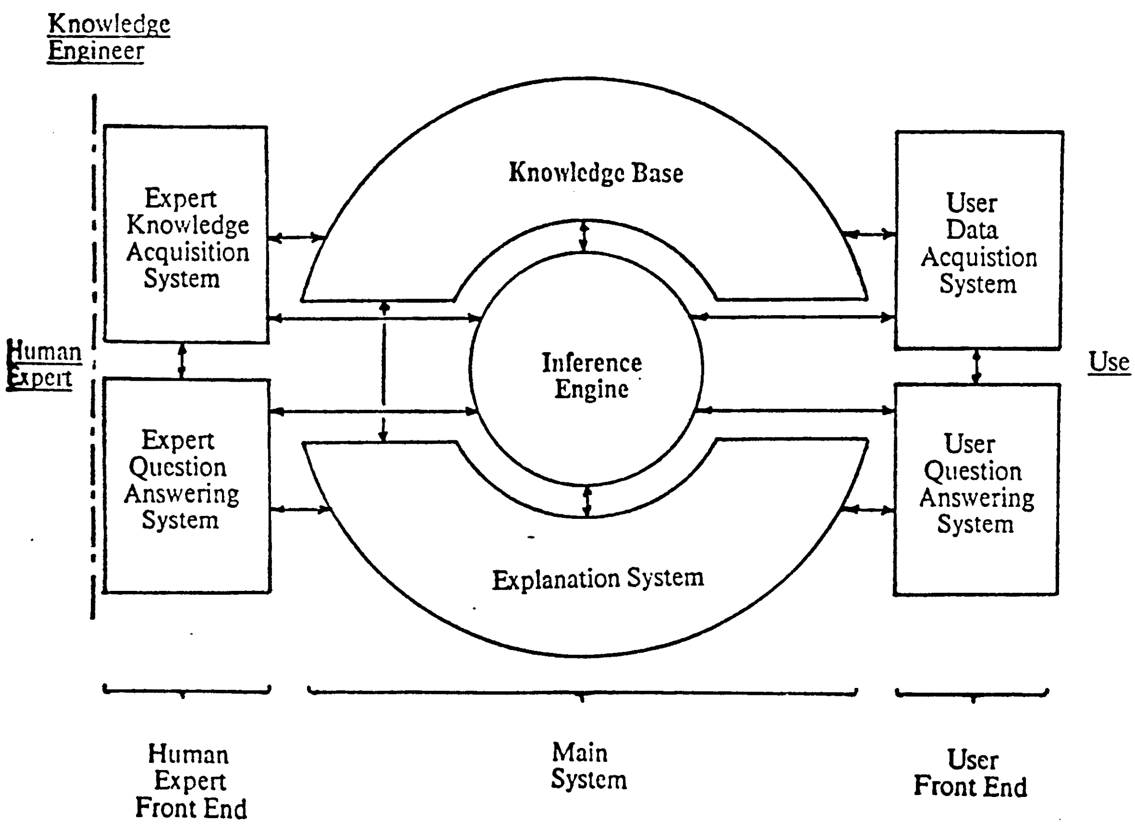
4.3.1 Architecture

Figure 1 and Figure 2 are representations of the possible complete architecture of today's expert systems.

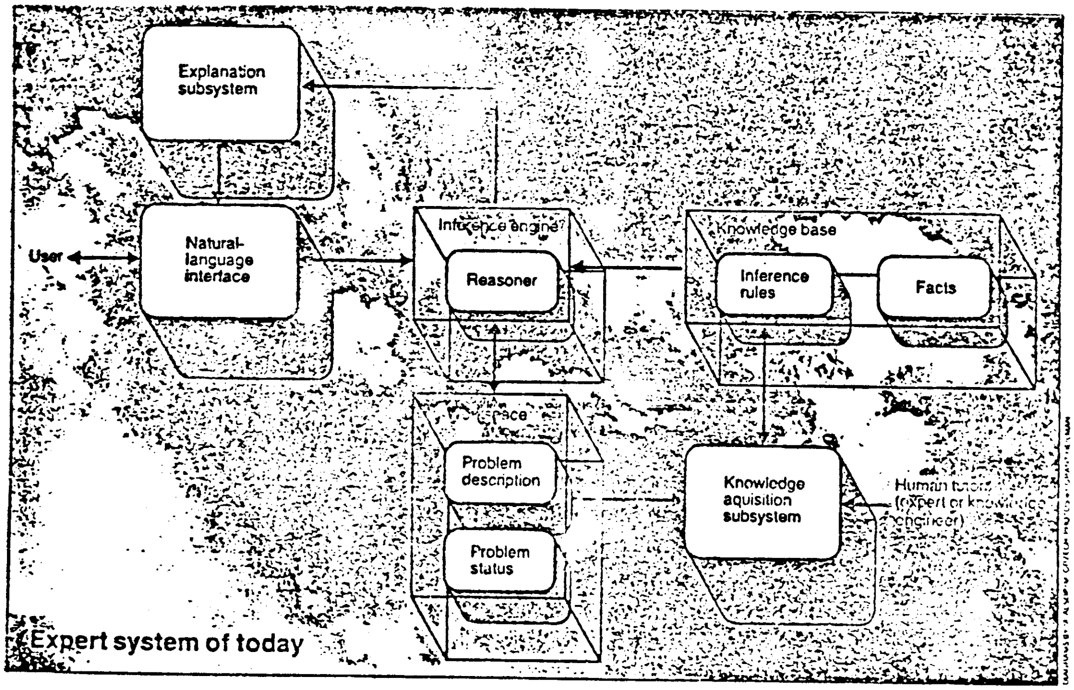
Source: Samuel Holtzman, "Artificial Intelligence,

Basic Concepts and an Introduction to Expert Systems

The Architecture of an Expert System



Paul Kinnucan, "Computers That Think Like Experts"
<u>High Technology</u> (January 1984) p.30



Some liberties are taken with these schematics, and no systems I know of are complete in all respects. The systems generators have the **potential** to be used to build systems with most of these components or all of them to some degree. Figure 1 is more of a flow chart than a schematic. Figure 2 breaks the flow chart down to higher resolution. Most of the components have been covered in depth in previous chapters, but a brief review and comment is appropriate.

4.3.1.1 Human Expert

> Knowledge comes, but wisdom lingers. It may not be difficult to store up in the mind a vast quantity of facts within a comparatively short time, but the ability to form judgments requires the severe discipline of hard work and the tempering heat of experience and maturity.
>
> -Calvin Coolidge

> An expert is one who knows more and more about less and less.
>
> -Nicholas Murray Butler

The human expert need not be computer familiar. Holtzman defines an expert as: "A **behavioral** definition, applicable to **humans** as well as to **computer systems**" he goes on to list attributes:

- Capable of using extensive **domain knowledge**
- **Reliable** (accurate and consistent)
- **Friendly** to the client/user

- **Adaptable** to a changing environment

- **Able to explain** her/his/its reasoning[159]

4.3.1.2 Explanation Systems

Expert	For the expert and knowledge engineer, explanation systems serve to document the logic and decision variable used to build the system. They also aid in debugging.
User	This is adding a "WHY" command to the standard "WHATIF" To enable the user to understand how and by what processes the system's recommendations were arrived at, as well as ensabling the user to ask for results under alternate scenarios. Also to justify requests for information.

4.3.1.3 Inference Engine

This was extensively reviewed earlier. It is the heart of the expert system.

4.3.1.4 User Interfaces

Also extensively reviewed earlier. These include natural language[160], and graphics.

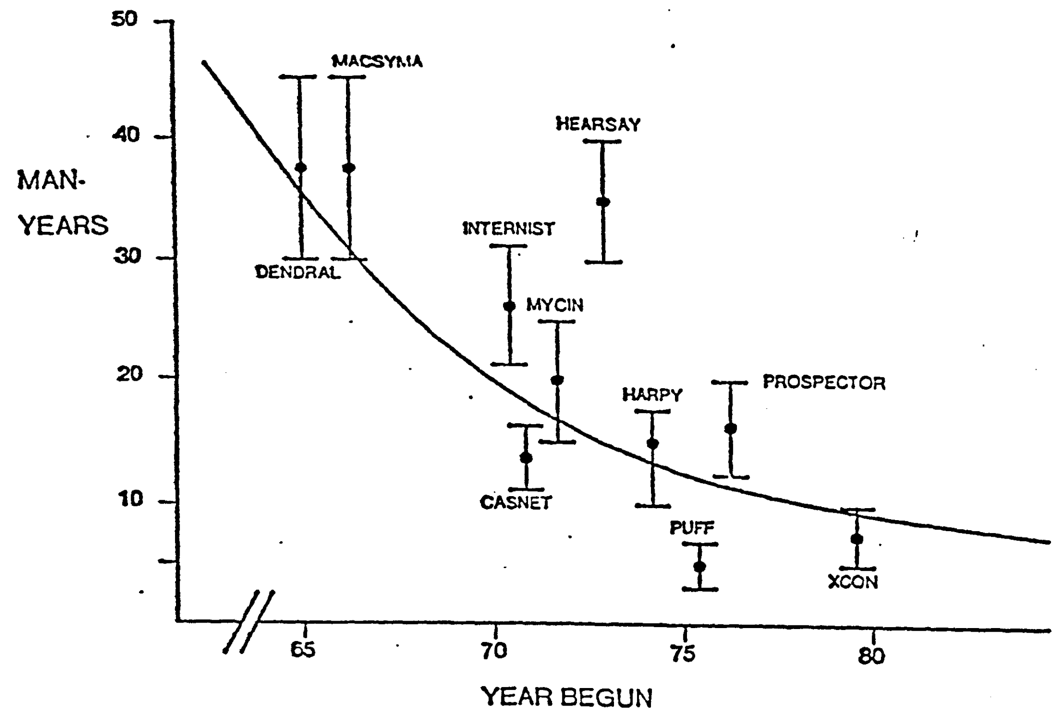
---

160. To the extent it is currently implemented, it is in a very limited form, highly specialized for the task at hand

4.3.2 Construction Process

Historically, the construction process was very time consuming. Figure 3 is a chart showing development time in man years for some well known expert systems.

Source:R. Davis "Expert systems: Where are we?
And Where Do We Go From Here?"
MIT Artificial Laboratory,
AI Memo No.
665, June, 1982.



While some progress has been made since Davis' research,[161] a good estimate of development time for a straightforward expert system is still three to four man years. When the emerging system generators begin to reach their potential, this may be cut to two to three manyears, more in line with conventional applications development for similarly complex tasks.

4.3.2.1 Knowledge Engineer

> Wisdom is the principle thing; therefore get wisdom: and with all thy getting get understanding.
>
> The Bible, Proverbs 4:7

A knowledge engineer is the opposite of an expert. He is computer literate, but domain ignorant.[162] His job is to carefully analyze, categorize and then codify the expert's knowledge, procedures/deterministic methods (algorithms, formulae, etc.) in the knowledge base in the form of data, rules, and frames. The knowledge engineer does this through interviews, simulation, and careful process analysis of how the expert goes about his task. The knowledge acquisition process takes into account:

- *The nature of the decision problem being dealt with.

  * What processes are involved in the problem solution.

---

162. It is usually better that he is so. A knowledge engineer with too much understanding of the domain may bias and filter the expert's knowledge during the process of encoding.

* What are the constraints on these processes

* What is given and what is inferred

The procedure employed as a basic problem solver.

* What strategies are employed

* What subtasks can be identified

* How are the objects in the domain related

* What is the flow of information

The allocation of responsibility to the user and the computer.

- What knowledge is needed to solve and what is needed to verify the solution.

- Resources required to acquire the knowledge, implement the system, and test it (i.e. time, computer facilities, and money).[163]

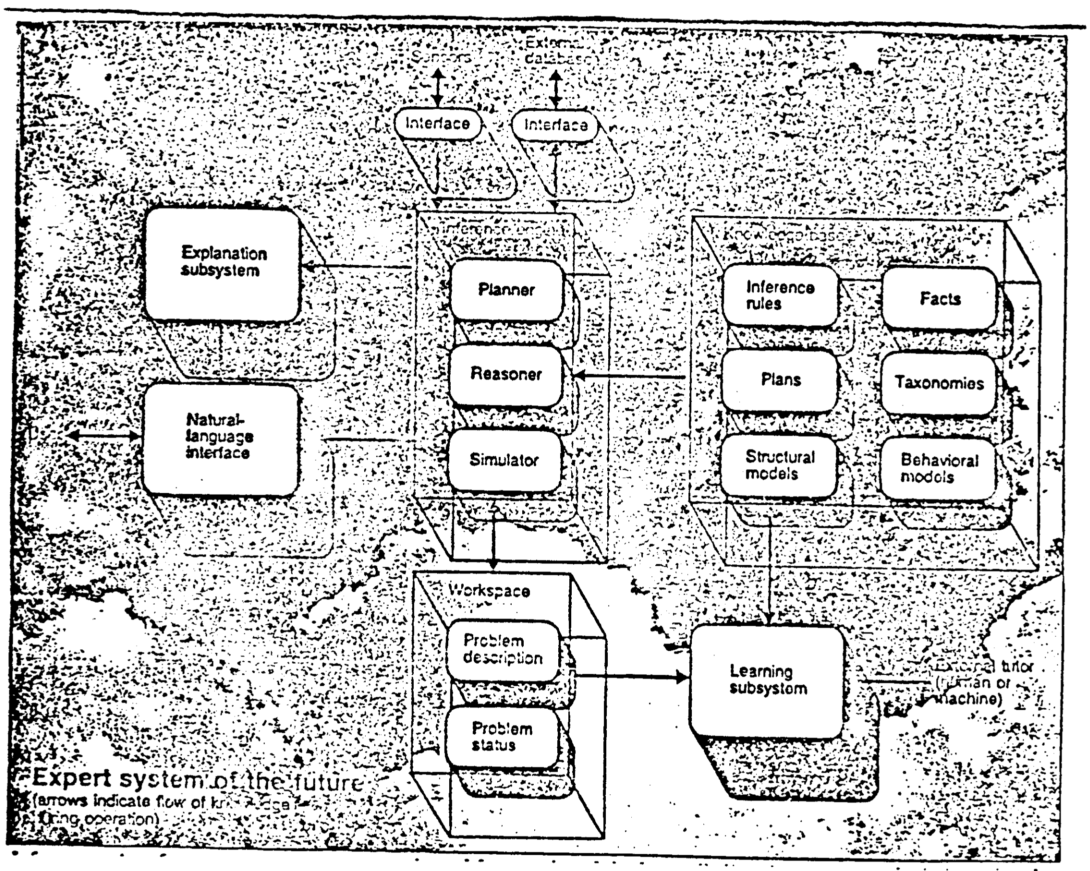
4.3.2.2 Knowledge Acquisition Systems

Knowledge acquisition systems are tools for inputing knowledge to the expert system. Now they are systems builders tools, in the future this process should be able to be automated. Currently, knowledge acquisition systems, from the standpoint of the user, basically just gather data and input criteria. They can't really **teach** the system without a knowledge engineer.

4.3.3 Future Enhancements

Figure 4 is a representation of the enhanced expert systems we should be able to expect to see by the end of this decade. It is a very stylized representation and should be only viewed as a set of directional goals. In order to accomplish these goals, which are briefly discussed below, major advances in Artificial Intelligence software technology **and** major advances in hardware technology are needed. Appendix A is a set of excerpts for the previously referenced DARPA report which detail that agency's estimates of time and work needed to built advanced expert systems. It is interesting to note that DARPA in expecting orders of magnitude improvements in the hardware at each step of development in order to support the expected improvements in software.

Paul Kinnucan, "Computers That Think Like Experts" High Technology (January 1984) p. 31.



4.3.3.1 Learning Subsystems

These are automated intelligent knowledge acquisition systems. They will be able to learn directly from end users by. In effect, being an expert "knowledge engineer" system themselves. Such systems will evolve from today's knowledge acquisition systems. Today's knowledge acquisition systems are really just "smart" input systems to aid the knowledge engineer in inputing data to an expert system. They still require the knowledge engineer to structure the problem, heuristics and data. They have some limited capability to allow end users to update these things, but they cannot diagnos the proble at hand, or solicit information to complete the inference engine's requirements to handle it.

As an extension of this capability, future knowledge acquisition systems will be able to learn by interacting with external computer databases, applications programs, and other expert systems. Someday we can expect expert systems to converse directly and educate each other on their respective capabilities. This in another area where meta-knowledge will play a key role. The system will have to understand, as humans do, how it does its job in order to teach another system to do this.

4.3.3.2 More Powerful Inference Engines

Today's inference engines are pretty much limited to working with production rules and "facts". In the future, their knowledge bases will have knowledge (and meta-knowledge) of Plans, Taxonomies, Structural models, and behavioral models. This will place them much closer to emulating true cognition. The inference engine engines themselves will have meta-knowledge and thus be able to plan their approach to problem solving as a function of the situation they are presented with. The same type of problem may be solvable through different methods, depending on what is known, what is required, and an assessment of pitfalls of the available approaches in each unique situation (i.e., choosing a method which may lead to a combinational explosion, or very lengthy computing time without a commensurate gain in accuracy or benefit). The inference engine will also be able to know when it needs more information than the user can provide, determine where to seek it, and then communicate appropriately with external programs. For some applications, such as a process control or monitoring situation, it will be able to accept direct sensory input.

4.3.3.3 Natural Language

By this we mean the type of natural language discussed earlier in this chapter. Natural language may be viewed as a continuous goal, where succeeding generations of expert systems, incorporating the latest research and development, slowly approach human capabilities.

Chapter 5

Prospects For Commercialization

The time has come, the walrus said,
To talk of many things:
Of shoes—and ships—and sealing wax—
Of cabbages—and kings—
And why the sea is boiling hot—
And whether pigs have wings.

-Lewis Carroll,
The Walrus and the Carpenter

Lewis Carroll would have been intrigued by Artificial Intelligence. He was a mathematician and logician, one of the very brotherhood of modern Prometheuses who are trying to steal nature's cognition and give it to machines. I don't believe that his professional descendants will ever entirely succeed. This is undoubtably in their favor, for if I recall; Zeus chained Prometheus to a mountain top and set a vulture to eating his liver out for all eternity. Artificial Intelligence is out of the nursery any ready to go to work. The challenge to the commercial software world is what to do with it. Eventually, almost all computer programs will incorporate AI features. For the nonce, the technology is too primitive and expensive to implement for universal use. So it is best used where it most complements our Natural

Intelligence.

5.1 Natural vs Artificial Intelligence

Don Kosy of Carnegie Mellon University, one of the leading research centers of AI research contrasts Natural Intelligence and Artificial Intelligence thus:

Natural Intelligence is:

Perishable — Brain researchers say we remember everything we are ever exposed to; but mercifully banish much of it from our conscious memories. Sort of a mental garbage collection process to free our minds of information which some non-cognitive arbitrator deems unlikely to be of use to us again.

Difficult to Transfer — Our knowledge is built up slowly, layer by layer. We must learn to add before we can do algebra, and algebra before calculus. Teaching another our expertise is a duplication of our own education. It can't be done en mass; and not every potential student has the aptitude or is interested.

Expensive — Education is costly in time and specie. As before, replicating it is equally so.

Erratic — Solving a problem may require multiple expertises. We can't all be experts at everything.

Difficult to Reproduce — Each person's knowledge results from a unique combination of aptitude, education and experience. It is literally impossible to duplicate a person's life, and therefore, the exact breadth, depth and context of his knowledge.

Artificial Intelligence is:

Permanent	Once taught, the system never forgets, or become incapable of recall.
Easy to Duplicate	Copy the code, copy the expert.
Inexpensive	Tape is cheap.
Comprehensive	Limited only by disk space.
Easy to Document	The user can know the why and how of the systems action.

Kosy's has highlighted the points of divergence between natural and artificial intelligence. Extreme proponents of AI might argue that only time creates the differentiation. That is, given enough time[164] researchers will be able to eliminate these points as meaningful differences. Perhaps. But for now, Kosy's classification suggests some directions for potential developers of expert systems to consider.

The primary goal should be to focus on tasks where there is specific domain knowledge available and where there are well proven algorithmic and heuristic methods for solution. It is no coincidence that researchers have chosen such tasks for the initial implementations of expert systems. These have tended to be diagnostic or synthetic tasks. PUFF, which diagnoses cardio-pulmonary problem is an example of the former; XCON, which takes customer requirements and creates computer configurations which will satify

---

164. And, one presumes, money.

them is an example of the latter. Appendix A contains these and other examples.

5.1.1 Appropriate Tasks

Many tasks do not require Artificial Intelligence to be useful and usable. Beyond usefulness, there are well developed "user friendly" interfaces which work perfectly well for a wide variety of applications where computer and human meet. Examples of such tasks and their interfaces:

ATMs	The automatic bank teller machines work comprehensively and efficiently for routine bank transactions. They use hierarchical menues, graphic displays, and touch button or touch screens. Derivatives of ATMs include automatic ticket dispensers at airport, and traveler's check dispensers.
Personal Software	Much of today's personal software uses color, help screens, hierarchical menues and status lines to guide and inform the user. The newest generation of personal computers offers graphic interfaces where much information is conveyed through icons. Mechanical interface devices such as mice and touch screens simplify interaction.

However, these examples are still conventional at their heart. They still require the user to understand and be proficient in the underlying domain knowledge and procedures and direct the interaction. Artificial Intelligence takes the opposite task; assuming the user is not able to direct the solution to his own problem.

The best short term prospects for expert systems meet these criteria:

- **Difficult tasks.** The nature of the users needs in inherently complicated and requires a great deal of a priori knowledge, experience, judgement, and technical proficiency.

- **Lack of Human Experts.** Supply and demand. Simply, there is more call for help than there are people trained and experienced enough to go around. Scarcity implies expense, so a large part of the [potential] market is excluded from assistance by price.

- **Lots of Data.** Tasks which require detailed or difficult analysis of lots of data, where the analysis required heuristics and judgement, not just mathematical manipulation.

- **Unusual Locales.** In accordance with any or all of the preceding, where the job to be done is remotely or inconveniently located. Such as diagnosing drilling problems on an off shore oil rig.

- **Substantial Economic Payoff.** Where expert performance at a task will yield substantial returns. Such as the Telephone Company's's ACE system which diagnosis service problems, devises a repair/preventative maintenance plan, and schedules work orders for the next day.

- **No Algorithmic Solutions.** Problems which are solvable only through procedural rules. Examples include problems where conventional approaches become combinationally explosive, such as chess; or where there is no purely deterministic solution, such as interpreting oil well wire line data; or where the problem is best addressed through heuristic search, such as medical diagnosis. Caveat: There may be no algorithmic solution, but there does have to be a set of stable rules.

- **Unskilled Users.**

- **Need to Learn From Experience.** Problems where the utility of the system results from the experience it gains over time. Again, Chess programs which learn from their play are a good example.

- **Instructional.** Expert systems have the virtue of being able to explain the how and why of their actions. Expanded, this also makes them ideal for teaching applications where they can track the progress of the student and tailor further instruction accordingly.

An important consideration which runs horizontally through these points is that the difference between the expert and the layman should be great and the potential payoff large. Therefore the would be much to be gained by abstracting the expert, rather than [attempting] to train the layman. For example, it has been statistically proven, that over the medium term[165], over 80% of the professional investment councelors perform **worse** than the average of the Standard and Poor' "500". There would seem to be little benefit to "expert systemizing" their knowledge and heuristics. On the other hand, experts specializing in interpreting wire line and seismic data obtained during oil exploration have significant greater success in locating hydrocarbon

---

165. Ten years or so.

deposits than inexperienced geologists. The payoffs for better interpretation are enormous. Succesful systems of this type are listed in Appendix A.

No attempt is made to priorize these criteria. The decision to build an expert system is properly based on a weighting of these factors, the lack of alternative ways of addressing it, and the monetary payoff. One or more of these criteria may suffice to justify the system.

5.2 An Expert System Prescription

The goal of this thesis is to assess the current potential for the commercialization of Artificial Intelligence technology. More specifically, expert systems as the part of that technology which appears to have the greatest current commercial possibilities.

Consumer product new product development work utilizes a techniques called focus groups and depth interviews. Focus groups entail assembling six to ten potential of current users of a product, which may or may not yet exist,[166] and having them participate in a mutual dialogue to explore their needs, complaints, and desires. The dialogue is led by a professional researcher who gently directs the conversation to relevant topics. The greatest utility from focus groups lies in the free flow of ideas and

---

166. If it doesn't exist, the moderator will expose the concept through mock-ups, slides, and other audio-visual aids.

opinions. The researcher later diagnoses useful information from the session.

Depth interviews are one on one sessions between a user or prospect, and a researcher. The interview is usually highly structured as the goal is to probe deeply into the interviewee's thinking and opinions on the matter.

I have conducted many interviews with potential purchasers of expert systems (for tasks to be discussed in the following section) in both focus group and depth interview settings; as well as with researchers and those in companies attempting to create products for sale. One of my goals was to try to ascertain what the operational attributes of a commercially desirable expert system would be. The following is a synthesis of what appeared to be most important to potentials purchasers of expert systems.

5.2.1 System Attributes

5.2.1.1 WAG (sic)

WAG[167] stands for **W**on't **A**ccept **G**arbage. The system should be smart enough to reject input which is highly suspect in the domain. WAG capability would also make assumption checks, and consistency checks. When the system completed its work, it would also review its own conclusions and check them for reasonableness and hidden implications and pitfalls which

---

167. A term coined by Professor Stewart Myers of MIT

could result from following the advice. For instance, if the a financial expert system requests a target rate of return from the user and is given "40%", it should question the reasonableness of the rate based on its knowledge of historical rates of returns. Another example would be users who input inconsistent data, such as preparing a financial forecast and assuming costs and revenue are projectable in constant dollars.

Further, it should discuss the situation with the user and educate him, where needed, to construct a more likely scenario. The system should also be able to step in when the user is on unfamiliar turf, and default to a "domain standard" scenario, and then notify the user and educate him if he so desires.

5.2.1.2 Normal Returns File

One of the key challanges for sellers of expert systems will be to win the user's confidence that they can depend on the system's recommendations. To that end, it is critical that the system can explain and document its action for the user. That user may need to provide such justification to a superior or co-worker in order to gain their approval. One aspect of obtaining credibility for recommendations is providing data which shows how the recommendation (or "answer" in the case of some problems) stacks up against the world of possibilities.

For example, if the system recommends that the company increase its debt load to35% of capitalization, the user will want to know what the mean and distribution is for debt load percentages in similar industries. This capability is sort of a reverse type of "WAG". Just as the system won't accept unreasonable input and commands for the domain, the user will want to know hoe the system's output stacks up against outside norms. Properly implemented, the normal returns file can even be a primary motivation for purchasing an expert system.

5.2.1.3 Alternative Methodologies

The system should explore alternative methodologies for solving a problem, where appropriate; and present the results for consideration and comparison, along with appropriate commentary and recommendations. It should be able to consider structural as well as parametric alternatives. Structural alternatives would be to add or subtract variables, parametric alternatives would be to change a coefficient and check for effect.[168]

5.2.1.4 Meta-Knowledge (again)

In this context, Meta-knowledge means that the system should have a model of the corporation (or any superset entity) underlying its local analysis. Thus as the system works with a user on a specific problem, it can add its knowledge of the larger set to the user's input to frame better recommendations. For example, knowing the corporate tax situation so as to

---

168. A type of automatic sensitivity testing.

better advise branch offices on lease/buy decisions.[169] Without such knowledge, the finest financial expert system might generate the wrong recommendation. This knowledge would also be useful to train or update the user on a larger set than he might otherwise have [current] knowledge of.

5.2.1.5 Cross Disciplinary Knowledge

Cross disciplinary knowledge is really a facet of assumption and consistency checking. It means that the system can provide assistance to the user on matters out of the domain the user is working in, but which may prove useful. For instance, many operations research techniques have utility for financial management, but few financial managers are familiar with operations management. The expert system should be able to cross these bounds when it sees similarities.

5.2.1.6 Learning

The system should be able to learn from the user through dialogue, teach by example, and extract knowledge through observation of the users' interaction.. There should also be safeguards on who can teach the system what. The knowledge engineer should be able to be eliminated, and the system should deal directly with the experts. One amusing suggestion was

---

169. A corporation paying no taxes should lease rather than buy. It can't use the ITC and depreciation from ownership, so it is better off to trade those deductions to a financial institution for a lower interest rate on its financing.

that one should be able to input a text or book into the system and instruct it to "learn about it"!

5.2.2 User Interface

Natural language did not rate as highly as one might think. This seemed to be due to disappointments many had had using products which **claimed** to be "English", but were subject to all the limitations discussed earlier. There was also a desire to have more of the information "pre-digested", and presented in a more efficient format that straight text. It will be important for the system to learn appropriate display techniques for various types of information. For example, what should be displayed graphically (and what the appropriate type of graph is, bar, pie, etc.), in chart, text, or some combination of forms. Key user interface criteria:

- Graphics. A strong propensity to have information presented graphically, and to interact with the system symbolically, as much as possible.

- Interactivity. The system must be highly interactive, and conversational in nature. Response time must be extremely rapid, with keystroke return being the goal.

- Specific user familiarity. The system should understand the level of proficiency of the user, and his going in assumptions and expectations.

5.3 Generic Product Categories

The terms "expert system" or "knowledge engineering" are quite general in nature. Hayes-Roth, et al have complied the following taxonomy for types of expert systems:[170]

Interpretation	Inferring situation descriptions from sensor data.
Prediction	Inferring likely consequences of given situations.
Diagnosis	Inferring system malfunctions from observable.
Planning	Designing actions.
Monitoring	Comparing observations to plan vulnerabilities.
Debugging	Prescribing remedies for malfunctions.
Repair	Executing a plan to administer a prescribed remedy.
Instruction	Diagnosing, predicting, repairing, and monitoring student behavior.
Control	Interpreting, predicting, repairing, and monitoring systems behavior.

5.4 Commercial Target Markets

> Grace is given of God, but knowledge is bought in the market.
>
> -Arthur Hugh Clough

Artificial Intelligence is a au currant topic. Not many who aren't in the field are aware of have active research and development projects. Some have been working in the area over ten years, other only within the past two or three; but the level of activity is high and is getting higher literally by the day. When the American Association For Artificial Intelligence held its first convention three years ago there were about 350 attendees, last year there were 3500. Appendix B contains a partial list of World Wide AI activities, and Appendix C is a partial list of selected experimental and operational expert systems. There is no attempt to be comprehensive, rather the notion is to give the reader a flavor of the level of activity.

As the technology diffuses through industry, we may expect to see many commercially purchasable expert systems. The following is a list of some of the areas I see as most promising:

- Systems Generators
- Equipment Fault Diagnosis
- Intelligent Interfaces. Particularly to DBMSs and widely used applications programs.
- Robotics
- Process Control
- Decision Support

- Medical Diagnosis and Prescription

- EDP Auditing

- Legal Counseling

In summary, I feel that the most promising areas for short term development are those which are primarily diagnostic and synthetic. These are production rule based applications and that part of the inference engine is well understood and developed. Humans are easily made at home with rules and such reasoning processes, this will make the knowledge engineer's job much easier and therefore shorten development times. Such applications are also well suited for verification. They can be pitted against the experts in well defined tasks, and their performance measured.

Chapter 6

Quo Vado? A Look Ahead To Adolescence

> We should all be concerned about the future because we will have to spend the rest of our lives there.
>
> -Charles Kettering

Artificial Intelligence is a many splendored thing. Most areas of computer hardware and systems software development are populated with dyed-in-the wool computer scientists, hardware specialists, systems analysts and programmers. While the general field of computer science is broad, people tend to specialize in narrow areas. Some design chips, some whole computers. Chip makers specialize in CPUs, memories or custom chips; computer designers choose supercomputers, mainframes, minis or micros. There are those who specialize in compiler designs, others just in operating systems. At the programming level, even experts often limit themselves to one or two high level languages.

Artificial Intelligence really transcends computer science. It deals with the way humans operate as much as the way computers do. The key research in the field is concerned with understanding how humans think, and then trying to emulate that in machinery. It is not "trying to make machines think". As a result, one meets the most interesting people along the way. Major contributors include computer scientists, psychologists, linguists, physiologists, grammarians, lexicographers, mathematicians, statisticians and even the odd philosopher or two. Research in all these areas has combined and conflicted to create the morphology and tools examined in the previous chapters. The purpose of this thesis is to set the framework for looking at what we can expect the Artificial Intelligence systems of today to do, and what we should look forward to them doing in the future.

6.1 Man, Golem, Responsibility and Ethics

> The future offers very little hope for those who expect that our new mechanical slaves will offer us a world in which we may rest from thinking. Help us they may, but at the cost of supreme demands upon our honesty and intelligence. The world of the future will be an ever more demanding struggle against the limitations of our intelligence, not a comfortable hammock in which we can lie down to be waited upon by our robot slaves.
>
> -Norbert Wiener,
> God and Golem, Inc.

Most of the more troubling questions surrounding the place of computers in society have centered around the computer's potential, real or perceived abilities to imitate the cognitive as well as arithmetic abilities of its creators. After forty years of development, computers are at the point of having an operational type of artificial intelligence capability. That the line between cognition and silicon may **apparently** be on the verge of being crossed, has highlighted moral and ethical questions as old as Golem and as new as 1984. These are issues of concern to both the developers and users of such technology. There is already sufficient concern about computer privacy to effect legislation regulating it. One can imaging the type of fears people will have about machines which may "judge" them in some way.

In the early days commentators were prone to a kind of supra-personification of computers, much of which has extended to this day. Computers were often referred to as "electronic brains" or "super-brains" or things along those lines. It was and, to a degree, still is fashionable to compare their capabilities with ours in terms how many operations of some sort the computer can do in a second. The quatrain usually begins ..."If ten men working 24 hours a day for ten thousand years......etc.".

Great expectations often lead to greater disillusionment. Humans proved not to be the biological equivalents of electrical circuitry. We are not, as Wiener attempted to show, highly sophisticated servomechanisms. Disciples of Pitts and McCulloch were disappointed by the failure their "Logical Calculus." Whether our intelligence and cognition is devinely inspired or not, there seems to be more to human thought than meets the engineer's eye. Not only was the problem far more complicated that the optimists believed, it was more complicated than the pessimists believed! As powerful as computers were, even superficially mimicing human thought required computational power that is only now becoming possible. While progress was made throughout the sixties and seventies towards machines which could meet the Turing test, the popular view of computers changed from Golem to data processing. Computer scientists and the business world scoffed at the notion of silicon alter egos with all the vehemence of reformed smokers and alcoholics towards cigarettes and John Barleycorn. Researchers in Artificial Intelligence were relegated to the same Elysian Field as alchemists.

6.1.1 Modern Times

Artificial Intelligence may have been residing in the Elysian Fields, but is was not dead. The lure was as as strong as ever among the faithful. As the decade of the eighties approach, popular thought was cautiously re-examining whether computers might think. More properly, thinking in Turing's terms rather than Wiener's and McCulloch's.

Manifold increases in computer "horsepower" and forty years of learning about programming techniques are starting to create programs which look deceptively cognizant. That would make the empiricists happy, they always believed perception is reality. But Plato is probably still laughing at our folly. Artificial Intelligence researchers wisely refrained from making the expansive claims of their predecessors, and have shied away from proclaiming their creations sentient being. Instead they have produced programs designed for specific, bounded, definable tasks which perform admirably. Commercial products started to emerge from these research attempts.

Twenty years of advancement in computer hardware and conventional software technology have brought us to the point where a whole generation has grown up with computers and come to depend on them for all sorts of utilities and services. Personal computers are appearing on the desks of poets for word processing as well as scientists for number crunching. We are thoroughly familiar with what computers can do. We try to keep our perspective about what they cannot. Most of the routine tasks have been successfully computerized. As the computer becomes a fixture in the home and office, as well as such places as hospitals, oil rigs and the classroom; a class of people not inclined to learn programming or deal with "dumb" machines is confronting the screen.

But how to harness this power for people not trained in it? How to make available techniques and information for people who need them but need guidance and aren't able to or inclined to write programs or understand complex mathematics. The popular and business press in once again focusing on Artificial Intelligence. Plus ca change, plus c'est la meme chose. But they are much more cautious than there forebearers. Scientists, vendors and commentators frame their claims and criticisms carefully. Nonetheless, progress has been made. With the future much closer at hand, it is worth considering not only how this technology can serve us but how we must guard against abuse.

There will be a great temptation to view expert and knowledge based systems as all knowing. Users may start to subordinate their own experience and judgement to the system. This is very dangerous because of the heuristic nature of these systems. People are accustomed to the determinism of a computer. If one programs a computer to add together a string of numbers, one usually doesn't question the answers. One never questions what **method** the computer used to do the addition, though there is a discrete algorithm by which the computer does addition.

Computers have posed moral and ethical questions since they started to appear in the late nineteen forties. There is a cliche now that computers don't make mistakes, people make mistakes. Because of their flawless arithmetic accuracy, people have a tendency to regard all computer output as equally sancrosect. This is, of course, fallacious; computer are programmed by people and therefore are subject to mistakes of direction. This mechanistic adherents to procedure is the source of society's amusing, and terrifying, stories about computers. Typically these stories are about computers which occasionally send out checks to average citizens for millions of dollars (errors in data input), or endlessly dunn people and destroy their credit ratings for unpaid balances of $00.01 (lack of heuristic rules and meta-knowledge). Like a ship with a damaged rudder, they will go in circles, or steer a course for the shoals, with all the determination of the inanimate, if that is how they have been set. With an expert system, the computer may well be wrong because its heuristics were wrong, or at least incomplete. Even the best computer chess programs lose games. Computers do not loose Tic-Tac-Toe games, because there is an algorithm which will always yield a win or, at worst, a tie.[171] But chess involves judgement and strategy.

6.1.2 Guarding Against HAL

Knowledge, like life, is a living process. Computers are processors, but they are not living. While Artificial Intelligence has created programs that learn from their experience, such learning is still at less than an infant's level. The knowledge once, entered in the computer, hardens with the rapidity of plaster of Paris, and holds its shape forevermore. It may not be so in the twenty first century when meta-knowledge is commonplace, but we must guard against such intellectual arteriosclerosis and becoming complacent with it.

Organizations must inculcate their cultures with the notion that expert systems are advisors, not seers. They are at least as fallible as their creators and may not see the [potential] error of their advice in every situation. Their knowledge will always be incomplete, as ours has always been. People must be encouraged to questions results from expert systems without fear of degradation, humiliation, or prejudice to their jobs if they feel something doesn't "add up". People must also remember that the machines serve people, not visa versa. Paraphrasing Wiener, intelligent systems will require **more** of us, not less.

---

171. The problem is also not combinationally explosive, so the algorithm can be run efficiently.

Lack of innovation and "excellence" is a popular subject of business literature. Nothing could stifle both more than an organization which views its expert systems as omnipotent. Most of us are fundamentally risk averse. An "officially" approved intelligent computer could provide a safe way to make decisions. It seems to me that it will be management's responsibility to assure that Pandora's box be opened only enough to see if we want whats inside but not enough to let out the misfortunes which attend the greedy and naive. Hans Christian Anderson wrote a fable, **The Emperor's New Clothes,** which dealt with what can happen when people subvert their own intelligence, observations and analysis. No one wanted to risk humiliation by pointing out the obvious. Worse, some of the people, flying in the face of reality, believed the obvious to be wrong because what they granted to be a higher authority, believed otherwise.

But he hasn't got anything on, a little child said.

It took the innocent to force the experienced to accept reality. We will need to keep our innocence about us when encountering Artificial Intelligence, expert system, and "smart" devices. It will pay to remember the word **Artificial** is the operative one. If we wish to view expert systems as an advisor, recommender, or decision maker; we must subject them to the same review we would apply to any executive. Infallibility is best left to God.

Expert systems can be built which learn from their experience. Ideally, when first installed, they embody the best deterministic and heuristic knowledge available. But who is to teach the teacher. Quis custodiet ipsos custodes. We must guard against such systems, especially in light of people's natural propensity to rely on them, from being bent to someone's selfish means. With conventional software it is possible to prevent the user (although not the really determined expert) from gaining access to the code itself. But in an expert system capable of learning, such access may not even be required to do good or damage. We must be careful to assure our expert systems receive the same quality of education as our children.

> If there is a hole in a' your coats,
> I rede you tent it;
> A chield's amang you takin' notes,
> And faith he'll prent it.
>
> -Robert Burns

Expert systems can be mandated by corporate fiat. It is possible to imbrue the expert system with heuristics, formula and judgmental criteria which appeal the corporate staff. Such as rejecting any project which does not yield a certain rate of return, or assumes a technological breakthrough. This might tempt them to mandate that the expert system be used as a judge, rather than an advisor, in the field as a way of extending their control and policies. One can imagine a memo sent to all concerned saying that no proposals for resources will be considered which have not passed the resource allocation expert system's's muster. Notice that this is significantly different from saying that such requests must be accompanied by the "opinion" of the expert system.

People at the receiving end of this might be tempted to create experiences for the system, or, if possible tamper with it directly, which would change the system's understanding of the world. As it took such contrived new experiences into account the system could modify its conclusions such that it would become the de facto manipulator of its users to according one person or group's ends. Thus plant managers might misdescribe (by, say, understatement) their plants' capacity to an expert system monitoring performance for finance and marketing. This, so the plant managers, always held something "up" their sleeve for emergencies, and so they appear to always be operating near capacity.

> Virtue itself turns vice, being misapplied; And vice sometime's by action dignified.
>
> -Shakespeare,
> Romeo and Juliet, II, iii, 21

Obviously such dangers present the corporation with Ulysses' dilemma of sailing between the Scylla and Charybdis. A careful course is necessary to steer people between the immanent disaster of either lure. The expert system must not be used as an implied endorsement for opinions, judgments and data not commonly agreed to be beneficial for all concerned. It is clear that the constitution and modifications of expert systems will become an important part of general corporate operating policy.

It is especially appropriate to examine the issue of "Big Brother" this year. Again, by their nature, Artificial Intelligence based products can be made to observe the user without the user's knowledge or consent. Not observe in the manner of dealing with the problem at hand, but to look for such things as:

- How long did it take [the user] to work through the problem, even with the expert system's help.
- How much guidance and explanation of what was done did the user need.
- Did the user "understand" what he was talking about, or was he over his head.
- Were his questions and approaches novel or routine.
- How much did he refer to or request other users experiences.

This list could go on. The use of such information could be very helpful or very destructive to the owner of the system and the user. In an educational application, such diagnostics could aid the teacher in structuring an appropriate program for the needs of the user. But the darker side is there as well. Should the manager judge his subordinate by how proficient the expert system perceives him to be. Further, there is the right of privacy. People's thoughts are generally no one else's business. We may all pursue lines of thinking which, prima facie, may seem silly or worse. The problem is also contextual. The expert system will, of course, only be able to record and analyze what is typed in. Many of our thoughts are intermediate in nature and, taken out of context, could be subject to gross misinterpretation.

In time, none of this will be lost on users. If they feel at risk because of either knowing that the system has the inherent capabilities to do these things, or knowing explicit examples where people have been subject to some type of review, even if that review was positive, they will shy away from using what could otherwise have been a very useful or critical tool.

Finally, there is the area which is farthest in the future, but in many ways, the most troubling. This is the computer's ability for pattern and speech recognition. There is a classic scene in Stanley Kubrick's movie **2001** which illustrates this point. "Hal" is the name of the spaceship's's computer. He is wired in to all aspects of control, communications, environment, etc. He has been programmed to think, reason, and act in a very human like fashion. So much so that the crew members, indeed the audience, quickly considers him to be human and speaks to him as Hal, he answers with first names as well. Concerned about some erratic behavior (!) Hal is exhibiting, two crew members sequester themselves in a soundproof room (knowing Hal has electronic "ears" through which they usually address "him") to discuss what to do. We see that Hal also has electronic "eyes" throughout the ship, one of which is observing the lip movements of the crew members thought a window of the room. Hal "reads" their lips (pattern, voice, and speech recognition), realizes they intend to deactivate [parts] of him, and subsequently kills one of them. The other crew member, after many travails, finally manages to "pull the plug" on

Hal's cognitive functions, and just leave the normal computer functions intact.

While Hal is not yet with us, machine vision and speech recognition are getting better every year. Factory robots already have pattern recognition ability, new Chrysler cars remind us (in a gentle female voice) to buckle our seat belts and when to change the oil. It is conceivable that they will be able to monitor our speech, read the mail on our desks, and drawn inferences from our behavior. This is not an attractive notion, but one which I believe, is possible by the turn of the century—a mere sixteen years from now.

What can we do? We can anticipate these problems and start to deal with them at the infant's emergence rather than waiting for adolescence when they have been arrested for delinquency. Every parent knows how difficult the teenage years can be. The difference for users of expert systems is that adults understand that teenagers, contrary to teenagers' belief, don't know everything. We take that into account when dealing with them. We must be sure to resist the temptation to grant computers what we don't grant humans. It is probably a good idea to have a vice president of Philosophy or Ethics appointed to give the company some non-sequential, associative and, occasionally, mystical thinking about the place of man and machine, and how the two should co-exist. A society advanced enough to create machines of this manner should be able to give them a good upbringing.

Human history is rich in philosophical and ethical thought. But from time out of mind, we have preached more good than we have practiced. Intelligent computers present us a new challenge. They have the capacity and promise to reshape our society, jobs, leisure time, and limitations. The time may come when the Office of Human Resources (formerly the Personnel Department) will have to be called the Office of Cognitive Employees. Perhaps managers will interview machines as well as people. Perhaps the managers will be machines. So we may as well start doing unto machines as we would have them do unto us.

Satis verborum. Enough of words, no more need be said.

FINIS

Chapter 7

Bibliography

Barr, Avrum. "Meta-Knowledge and Cognition." International Joint Conferences on AI 6 (1979).

Barr, Avrum, and Feigenbaum, Edward A. The Handbook Of Artificial Intelligence, Vol II. Stanford: HeurisTech Press, 1982.

Bates, Madeline, and Bobrow, Robert. "Natural Language Interfaces: Whats Here, Whats Coming, and Who Needs It." in Artificial Intelligence Apllications for Business, Edited by Walter Reitman. Norwood: Ablex Publishing, 1984.

Berekely, George. Three Dialogues. ed. by M. Turbayne, New York: Liberal Arts Press, 1954.

Berger, T. Rate Distortion Theory. Englewood CLiffs: Prentice Hall, 1971.

Boole, George. An Investigation of the Laws of Thought on Which are Founded the Mathenatical Theories of Logic and Probabilities. 1854.

Brightman, Harvey J., Schkade, Lawrence L., Schoech, Dick. "Managing The Decision Sciences Technology: A Case For Ethical Analysis?" Decision Sciences 12 (October 1981) :4

Burke, James. Connections. Boston: Little, Brown and Company, 1978.

Davis, R.; Buchanan, G.B.; and Shortliffe, E. "Production rules as a representation for a knowledge-based consultation program." Artificial Intelligence. 8 (1977) :15-45

DeLong, Howard. A Profile of Mathematical Logic. Reading: Addison-Wesley, 1970.

Doyle, J. "A model for deliberattion, action, and introspection" Ph.D. dissertation, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980.

Dreyfus, Hubert. What Computers Can't Do: A Critique of Artificial Reason. New York: Harper and Row, 1972.

Duda, R.O.; Hart, P.E.; and Nilsson, N.J. "Subjective Bayesian methods for rule-based inference systems." Proceedings og the AFIPS 1976 National Computer Conference.: 45., 1976.

Dunlop, John., ed. Automation: Implications For The Future. New York: Vintage Books, 1962.

Feigenbaum, Edward E., and McCorduck, Pamela. The Fifth Generation. Reading: Addison-Wesley, 1983.

Freidrich, Gerhardt. Digital Equipment Corporation, Boston, Massachusetts. Interview, 11 January 1984.

Gloess, Paul Y., Understanding Artificial Intelligence. Sherman Oaks: Alfred Publishing, 1981.

Hamming, R.W., "Error Detecting and Error Correcting Codes." Bell System Tech Journal 29 (1956).

Godel, Kurt. On Formally Undecidable Propositions. New York: Basic Books, 1962.

Gould, Stephen Jay. Ever Since Dariwn. New York: W.W. Norton & Company, 1977.

Hayes-Roth, B.; and Hayes-Roth, F., Cognitive Processes in Planning. Santa Monica: RAND Corporation, 1978.

Hodges, Andrew. Alan Turing: The Enigma. New York: Simon and Schuster, 1983.

Hofstadter, Douglas R. Godel, Escher, Bach. New York: Basic Books, 1979.

Holtzman, Samuel. "Artificial Intelligence; Basic Concepts and Introductions to Expert Systems." Presentation, Stanford University.

Kinnucan, Paul. "Computer That Think Like Experts." High Technology 4 (January 1984): 30-42.

Kleene, S.C. Mathematical Logic. New York: John Wiley and Sons, 1967.

Lewis, C.I.; and Langford, C.H. History of Symbolic Logic, in The World of Mathematics. J.R. Newman, editor. New York: Simon and Schuster, 1956.

Lovelace, Lady A.A., Notes Upon the "Sketch of the Analytic Engine Invented by Charles Babbage." reprinted in Morrison, P.; and B., Charles Babbage and Hid Calculating Engines. Geneva, 1842.

Mathews, M.; and Rosler, L., "A Graphical Language For Computer Sounds." in Music By Computers Edited by H. von Foerster and J.W. Beauchamp. New York: John Wiley, 1969.

McCarthy, J. "History of Lisp." SIGPLAN Notices. 13 (1978)

Nahm, Milton. Selections From Early Greek Philosophy. New York: F.S. Crofts, 1947.

Nagel, Ernest, and Newman, James. Godel's Proof. New York: New York University Press, 1958.

Neumann, John von. The computer and the Brain. New Have: Yale University Press, 1958.

Parkinson, C. Northcote. Parkinson's Law. New York:Ballantine Books, 1975.

Popkin, Richard A.; and Stroll, Avrum. Philosophy Made Simple. New York: Doubleday & Company, 1956.

Pople, Harry. "Knowledge Based Expert Systems: The Buy or Build Decision." in Artificial Intelligence Applications for Business., Edited by Walter Reitman. Norwood: Ablex Publishing, 1984.

Reid, C. Hilbert George Allen & Unwin; Springer Verlag, 1970.

Sagan, Carl. Broca's Brain. New York: Random House, 1974.

Shannon, Claude. "A Mathematical Theory of Communications." Bell System Tech. Journal (July 1948 and October 1948)

Shannon, Claude. "A Symbolic Analysis of Relay and Switching Circuits." Trans Amer. I.E.E.. 57 (1938)

Shortliffe, E.H., Computer Based Medical Consultation: MYCIN. New York: American Elsevier, 1976.

Simon, Herbert. The New Science of Management Decision. Englewood Cliffs: Prentice-Hall, 1965.

Shwartz, Steven. "Natural Language Processing in the Commercial World" in Artificial Intelligence Applications for Business., Edited by Walter Reitman. Norwood: Ablex Publishing, 1984.

Stefik, M. "Planning With Constraints." Ph.D dissertation, Stanford University, 1980.

Stevens, Al. Bolt, Beranek, and Newman, Cambridge, Massachusetts. Interview, 6 April, 1983.

Suppes, P. Introduction to Logic. Princeton: D. Van Nostrand Reinhold, 1957.

Turing, Alan, "On Computable Numbers, with an Application to the **Entscheidungsproblem**." Proc. Lond. Math. Soc. 2, 42 (1937)

Turing, Alan. "Computing Machinery and Intelligence." Mind Vol. LIX No. 236 (1950).

Laurance Urdang, ed., The Random House Dictionary of the English Language, College Edition. (New York: Random House, 1968.)

U.S. Dept. of Defense. Strategic Computing, New-Generation Computing Technology: A Strategic Plan for its Development and Application to Critical Problems in Defense. p. 7.

Webber, Bonnie,; and Finn, Tim., "In Response: Next Steps in Natural Language Interaction." in Artificial Intelligence Applications for Business., Edited by Walter Reitman. Norwood: Ablex Publishing, 1984.

Wiener, Norbert. Cybernetics Cambridge: MIT Press, 1961.

Williams, Chuck. "ART The Advanced Reasoning Tool, A Conceptual Overview." Explantory paper. Los Angelos: Inference Corporation.

Winston, Patrick. Artificial Intelligence. Reading: Addison-Wesley, 1979.

Woods, W.A. "A Personal View of Natural Language Understanding." SIGART Newsletter 61 (February 1977)

Appendix A

Examples of Expert Systems

The source for these charts in Feigenbaum and McCorduck's book **The Fifth Generation.**[172]

172. E. Feigenbaum and P. McCorduck, The Fifth Generation, (Reading: Addison-Wesley, 1983) pp. 244-249

SELECTED EXPERIMENTAL AND OPERATIONAL EXPERT SYSTEMS

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
Bioengineering	MOLGEN: Aids in planning experiments involving structural analysis and synthesis of DNA	Heuristic Programming Project, Stanford University
Chemical Industry	DENDRAL: Interprets data produced by mass spectrometers and determines not only a molecule's structure, but also its atomic constituents	Heuristic Programming Project, Stanford University
	SECS: Operational expert system to assist chemists in organic synthesis planning	University of California, Santa Cruz
Computer systems	DART: An experimental expert system for diagnosing computer system faults; used in field engineering	Heuristic Programming Project, Stanford University, and IBM
	R1 and XCON: Operational expert systems that configure VAX computer systems	Carnegie-Mellon University and Digital Equipment Corporation
	SPEAR: An expert system under development for analysis of computer error logs; used in field engineering	Digital Equipment Corporation
	XSEL: An extension of XCON that assists salespeople in selecting appropriate computer systems	Digital Equipment Corporation

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
	———: An experimental expert system for diagnosing VAX computer failures	M.I.T.
Computing	PROGRAMMER'S APPRENTICE: An expert system for assisting software construction and debugging	M.I.T.
	PSI: Composes simple computer programs based on English descriptions of the task to be performed	Kestrel Institute, Systems Control Technology
Education	GUIDON: An experimental intelligent computer-aided instruction (CAI) system that teaches the student by eliciting and correcting answers to a series of technical questions	Heuristic Programming Project, Stanford University
	———: An expert system under development that will teach computer languages to programmers	Computer Thought, Inc.
Engineering	EURISKO: An experimental expert system that learns by discovery; applied to designing new kinds of three-dimensional microelectronic circuits	Heuristic Programming Project, Stanford University
	KBVLSI: An experimental system to aid in the development of VLSI designs	Xerox Palo Alto Research Center and Stanford University
	SACON: An operational expert system that assists structural engineers in identifying the best analysis strategy for each problem	Heuristic Programming Project, Stanford University
	———: An expert system under development for nuclear power reactor management	Hitachi Energy Lab

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
	———: An expert system under development for diagnosing fabrication problems in integrated circuit manufacturing	Hitachi System Development Lab
General-purpose tools	AGE: A system that guides the development of expert systems involving hypothesis formation and information fusion	Heuristic Programming Project, Stanford University
	AL/X: A commerical expert system that assists diagnostic experts in encoding their knowledge of a scientific domain, thus generating a system able to exercise knowledge on their behalf; based on PROSPECTOR design	Intelligent Terminals, Ltd.
	EMYCIN: A basic inference system derived from MYCIN that is applicable to many fields: used in building PUFF, SACON, and many other systems	Heuristic Programming Project, Stanford University
	EXPERT: A basic inference system used in oil exploration and medical applications	Rutgers University
	KAS: An experimental knowledge acquisition system that creates, modifies, or deletes various kinds of rule networks to be represented in the PROSPECTOR system	SRI International
	KEPE: A commercially available knowledge representation system	IntelliGenetics, Inc.
	KS-300: A commercial basic inference system for industrial diagnostic and advising applications	Teknowledge, Inc.
	LOOPS: An experimental knowledge representation system used in KBVLSI	Xerox Palo Alto Research Center

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
	MRS: "Metalevel Representation System" for knowledge representation and problem-solving control	Heuristic Programming Project, Stanford University
	OPS: A basic inference system applicable to many fields; used for R1 and AIRPLAN	Carnegie-Mellon University
	ROSIE: A basic inference system applicable to many fields	RAND Corporation
	SAGE: A basic inference system applicable to many problems	SPL International
	TEIRESIAS: Transfers knowledge from a human expert to a system and guides the acquisition of new inference rules	Heuristic Programming Project, Stanford University
	UNITS: A knowledge representation system used in building MOLGEN and in conjunction with AGE	Heuristing Programming Project, Stanford University
Law	LDS: An experimental expert system that models the decision-making processes of lawyers and claims adjusters involved in product liability legislation	RAND Corporation
	TAXMAN: An experimental expert system that deals with rules implicit in tax laws and suggests a sequence of contractual arrangements that a company can use to attain its financial objectives	Rutgers University
Management science	KM-1: An experimental knowledge management system that attempts to integrate the capabilities of the data management system and knowledge base system	System Development Corporation

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
	RABBIT: An experimental system that helps the user formulate queries to a data base	Xerox Palo Alto Research Center
	———: An expert system under development for project risk assessment for large construction projects	Hitachi System Development Lab
	———: An expert system under development for cost estimation of steam boilers	Hitachi System Development Lab
Manufacturing	CALLISTO: An experimental system that models, monitors, schedules, and manages large projects	Robotics Institute, Carnegie-Mellon University
	ISIS: An experimental system used for job shop scheduling	Robotics Institute, Carnegie-Mellon University
Medicine	ABEL: An expert system for diagnosing acid/base electrolyte disorders	M.I.T.
	CADUCEUS: An expert system that does differential diagnosis in internal medicine	University of Pittsburgh
	CASNET: A causal network that associates treatments with various diagnostic hypotheses (such as the severity or progression of a disease); applied to glaucoma	Rutgers University
	MYCIN: An operational expert system that diagnoses meningitis and blood infections	Heuristic Programming Project, Stanford University
	ONCOCIN: An oncology protocol management system for cancer chemotherapy treatment	Heuristic Programming Project, Stanford University
	PUFF: An operational expert system that analyzes patient data to identify possible lung disorders	Heuristic Programming Project, Stanford University

DOMAIN	SYSTEM/DESCRIPTION	RESEARCH AND DEVELOPMENT ORGANIZATION
	VM: An expert system for monitoring patients in intensive care and advising about respiratory therapy	Heuristic Programming Project, Stanford University
Military	AIRPLAN: An expert system under development for air traffic movement planning around an aircraft carrier	Carnegie-Mellon University and U.S.S. *Carl Vinson*
	HASP/SIAP: An expert system for identification and tracking of ships using ocean sonar signals	Systems Control Technology, Inc., and Heuristic Programming Project, Stanford University
	TATR: An expert system for tactical air targeteering; uses ROSIE	RAND Corporation and U.S. Air Force
	———: Prototype expert system for analysis of strategic indicators and warnings	ESL, Inc., and Teknowledge, Inc.
	———: Prototype expert system for tactical battlefield communications analysis	ESL, Inc., and Teknowledge, Inc.
Resource exploration	DIPMETER ADVISOR: An expert system that analyzes information from oil well logs	Schlumberger
	DRILLING ADVISOR: An operational expert system for diagnosing oil well drilling problems and recommending corrective and preventive measures; uses KS-300	Teknowledge, Inc., for Elf-Aquitaine
	HYDRO: A computer consultation system for solving water resource problems	SRI International
	PROSPECTOR: An expert system that evaluates sites for potential mineral deposits	SRI International
	WAVES: An expert system that advises engineers on the use of seismic data analysis programs; for oil industry; uses KS-300	Teknowledge, Inc.

Appendix B

Worldwide Artificial Intelligence Activity[2]

2. Ibid., pp. 251-253

WORLDWIDE ARTIFICIAL INTELLIGENCE ACTIVITY

ORGANIZATION	LOCATION	APPLICATION AREA
AIDS	Mountain View, CA	Expert systems
Applied Expert Systems	Cambridge, MA	Financial expert systems
Artificial Intelligence Corp.	Waltham, MA	Natural language systems
Automatix, Inc.	Billerica, MA	Robotics and vision systems
Bell Laboratories	Murray Hill, NJ	Natural language and expert systems, data base interface
Boeing Co.	Seattle, WA	Robotics and process planning systems
Bolt Beranek & Newman, Inc.	Cambridge, MA	Natural language and instructional systems
Brattle Research Corp	Boston, MA	Financial expert systems, market survey
Carnegie-Mellon University	Pittsburgh, PA	Robotics, vision and process planning systems
Cognitive Systems, Inc.	New Haven, CT	Natural language systems
Columbia University	New York, NY	General AI
Computer Thought Corp.	Richardson, TX	Instructional systems
Daisy	Sunnyvale, CA	Expert systems and professional work station
Digital Equipment Corp.	Maynard, MA	Expert systems
Electrotechnical Laboratory	Tsukuba, Japan	Robotics and general AI

ORGANIZATION	LOCATION	APPLICATION AREA
Fairchild Camera & Instrument Corp.	Mountain View, CA	VLSI design and expert systems
Fujitsu-Fanuc Ltd.	Kawasaki, Japan	Fifth Generation computer
General Electric Co.	Schenectady, NY	Robotics, process planning and expert systems
General Motors Corp.	Detroit, MI	Robotics and vision systems
Hewlett-Packard Co.	Palo Alto, CA	Expert systems
Hitachi Ltd.	Tokyo, Japan	Fifth Generation computer
Honeywell, Inc.	Minneapolis, MN	Robotics systems
Hughes Aircraft Co.	Torrance, CA	—
Imperial College, London	London, England	General AI
IntelliGenetics, Inc.	Palo Alto, CA	Expert systems
Intelligent Software, Inc.	Van Nuys, CA	General AI
International Business Machines	Armonk, NY	Robotics and fault diagnosis systems, data base interface
Jaycor	Alexandria, VA	Expert systems
Kestrel Institute	Palo Alto, CA	Automated programming
Lisp Machines, Inc.	Cambridge, MA	Professional work station
Lockheed Electronics	Plainfield, NJ	Intelligent interface
Arthur D. Little	Cambridge, MA	Consulting
Machine Intelligence Corp.	Sunnyvale, CA	Robotics, vision and natural language systems
Martin Marietta Aerospace Co.	Denver, CO	Robotics systems
Massachusetts Institute of Technology	Cambridge, MA	Robotics and sensor systems, general AI
Mitre Corp.	Bedford, MA	Command control and decision support systems
Mitsubishi Electric Corp.	Tokyo, Japan	Fifth Generation computer
Nippon Electric Co. Ltd.	Tokyo, Japan	Fifth Generation computer
Nippon Telephone & Telegraph Corp.	Tokyo, Japan	Fifth Generation computer
Ohio State University	Columbus, OH	Robotics and general AI

ORGANIZATION	LOCATION	APPLICATION AREA
RAND Corp.	Santa Monica, CA	General AI
Rutgers University	New Brunswick, NJ	General AI
Schlumberger-Doll Research	Ridgefield, CT	Expert systems
Smart Systems Technology	Alexandria, VA	Instructional systems, AI tools
SRI International	Menlo Park, CA	Robotics and sensor systems, general AI
Stanford University	Stanford, CA	Robotics, vision and expert systems, VLSI design
Symantec	Palo Alto, CA	Natural language systems
Symbolics	Cambridge, MA	Professional work stations
Systems Control, Inc.	Palo Alto, CA	Expert systems
Teknowledge, Inc.	Palo Alto, CA	Expert systems
Texas Instruments	Dallas, TX	Instructional and robotics systems
Three Rivers Computer Corp.	Pittsburgh, PA	Professional work stations
TRW, Inc.	Cleveland, OH	Expert systems
United Technologies Corp.	Hartford, CT	General AI
University of Edinburgh	Edinburgh, Scotland	General AI
University of Illinois	Urbana, IL	Robotics and general AI
University of Marseilles	Marseilles, France	General AI
University of Massachusetts	Amherst, MA	Robotics and vision systems, general AI
University of Michigan	Ann Arbor, MI	Robotics and vision systems, general AI
University of Sussex	Sussex, England	General AI
Westinghouse Electric Corp.	Pittsburgh, PA	Robotics and expert systems

Appendix C

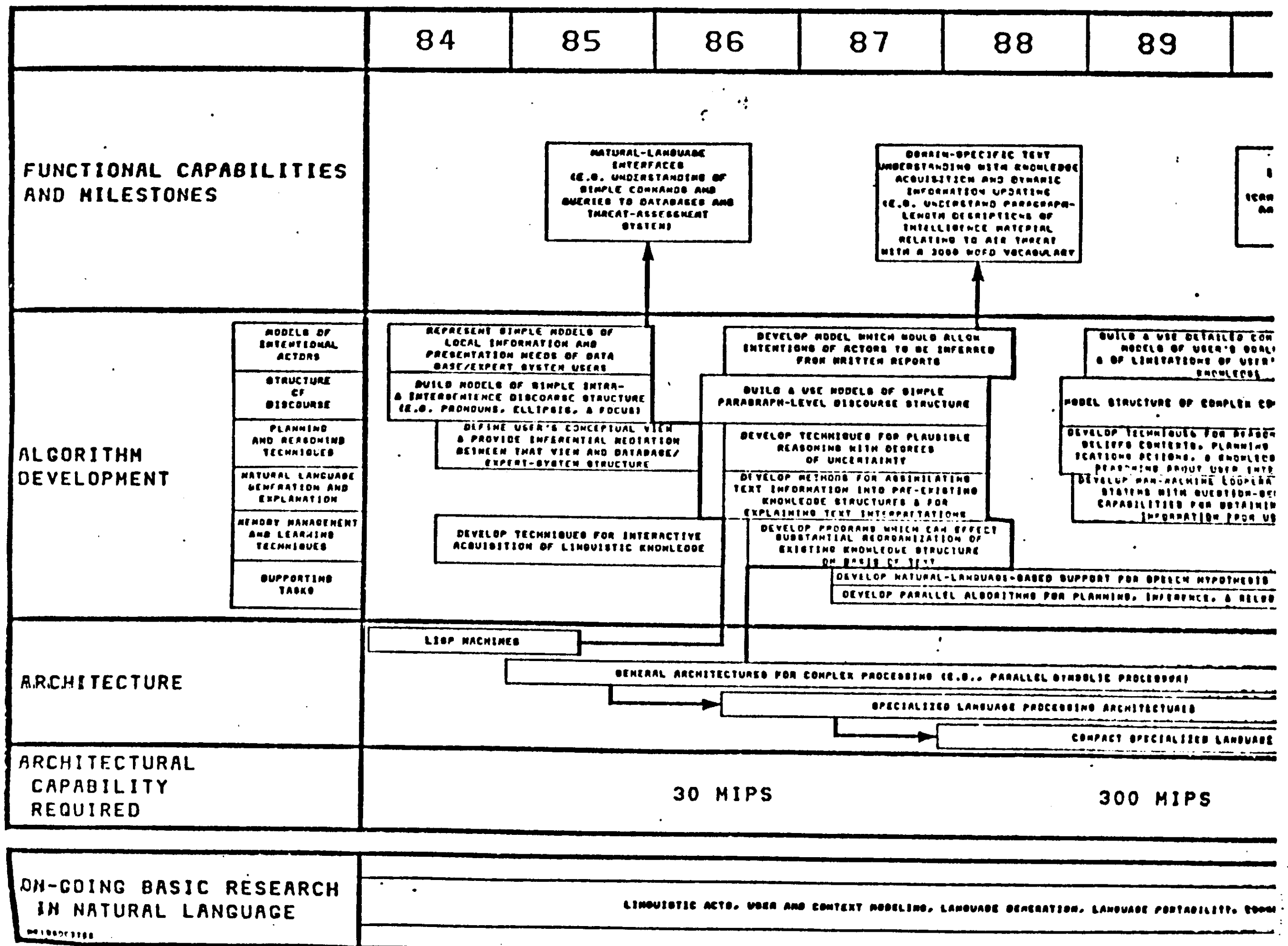
Expert Systems-Development Flow Charts (Advanced)

The source for these charts is **Strategic Computing**, a report from the Defense Advanced Research Projects Agency.[3]

---

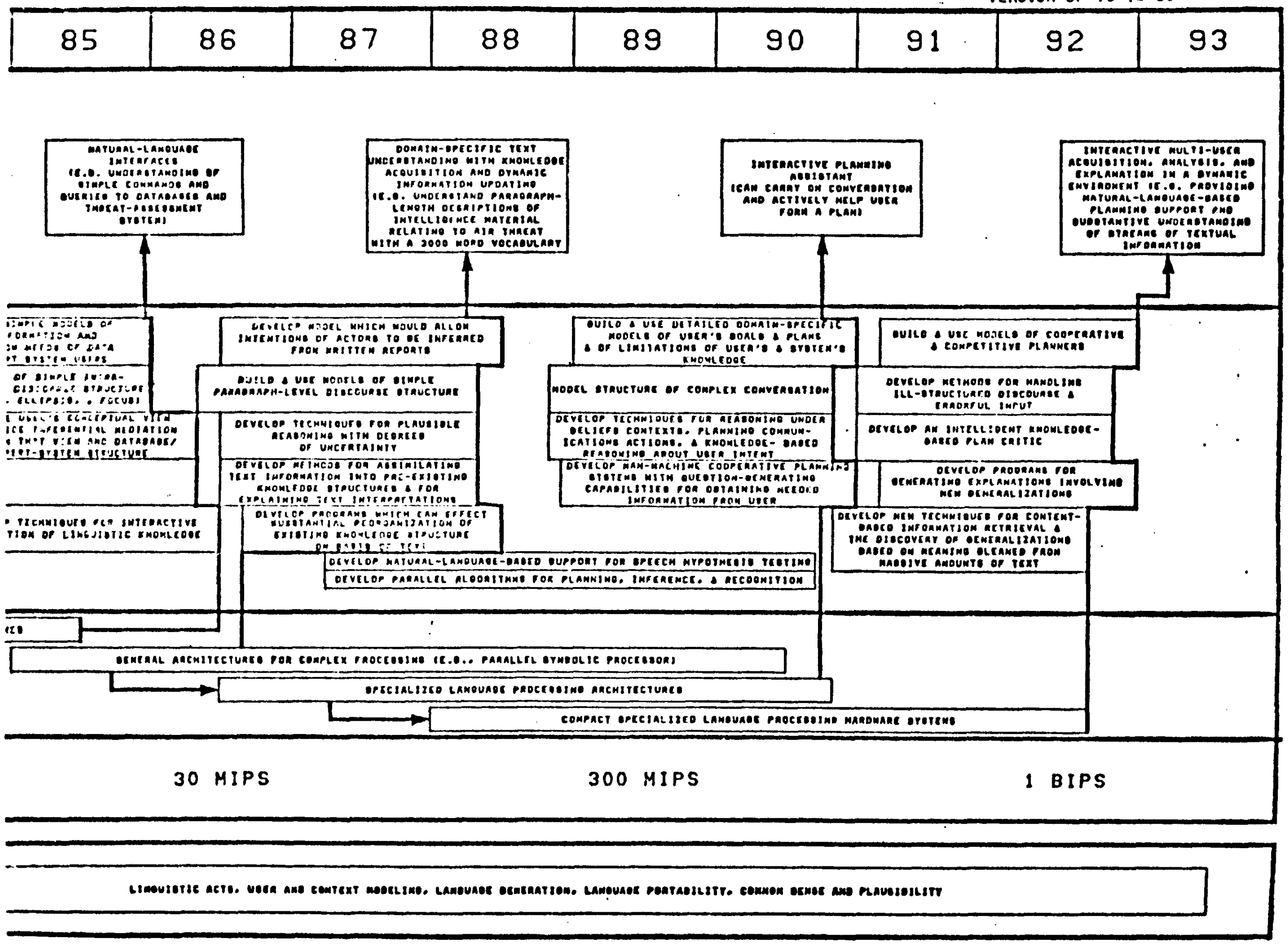
3. Strategic Computing (Defense Advanced Research Projects Agency, [1983])

NATURAL LANGUAGE SUBSYST



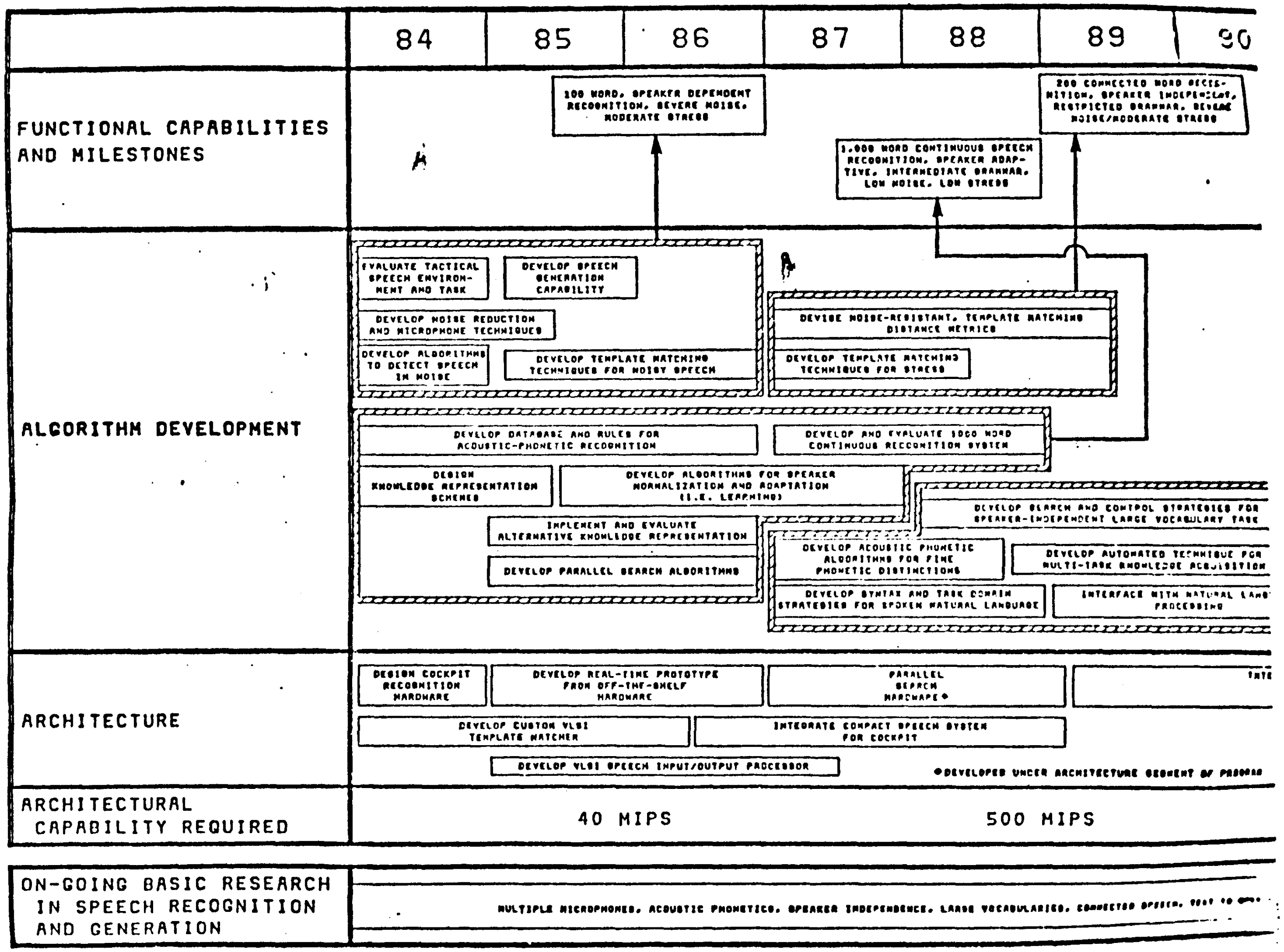
-URAL LANGUAGE SUBSYSTEMS

VERSION OF 10-18-83 II.1.3



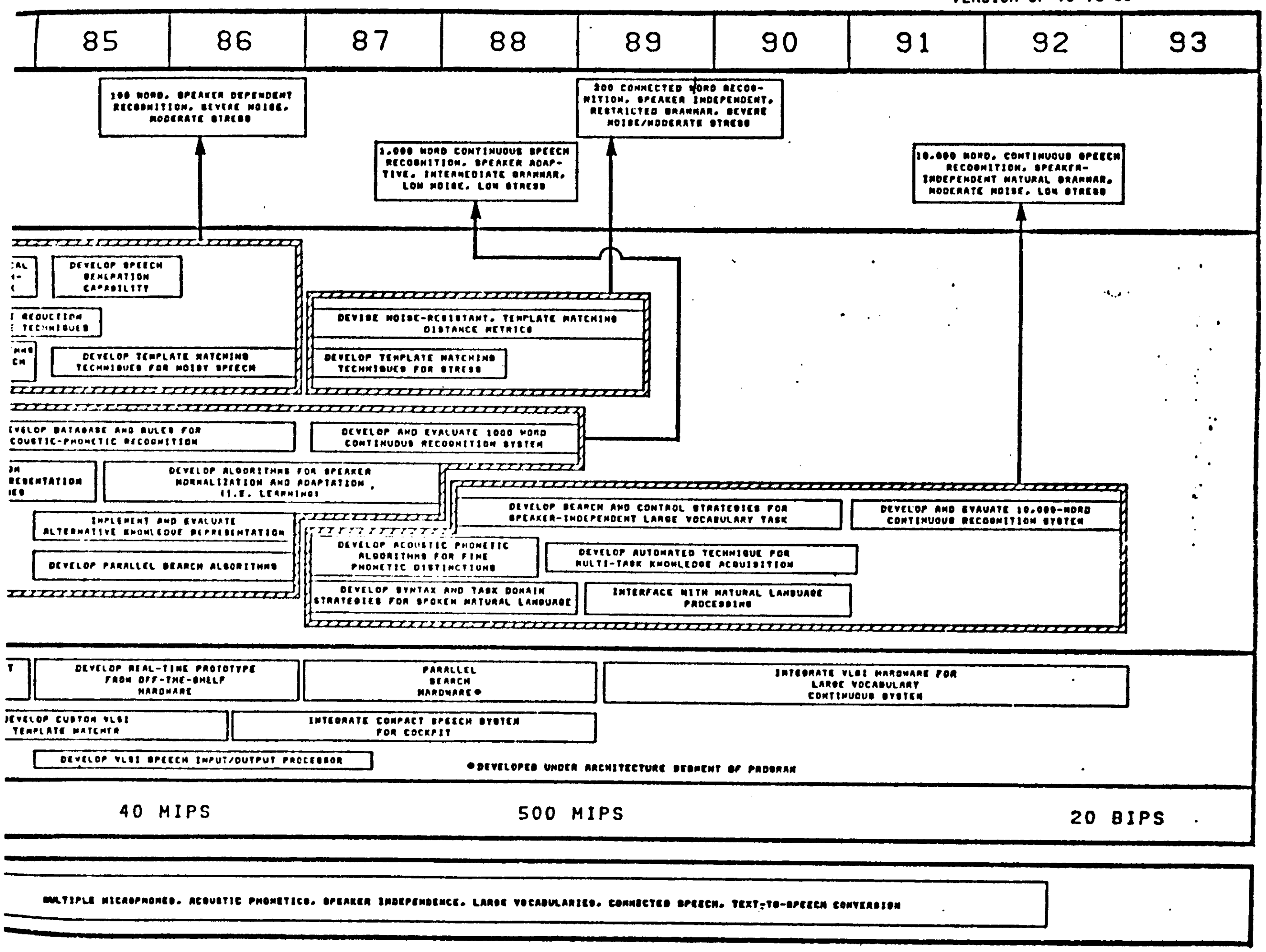
- 202 -

SPEECH SUBSYSTEMS

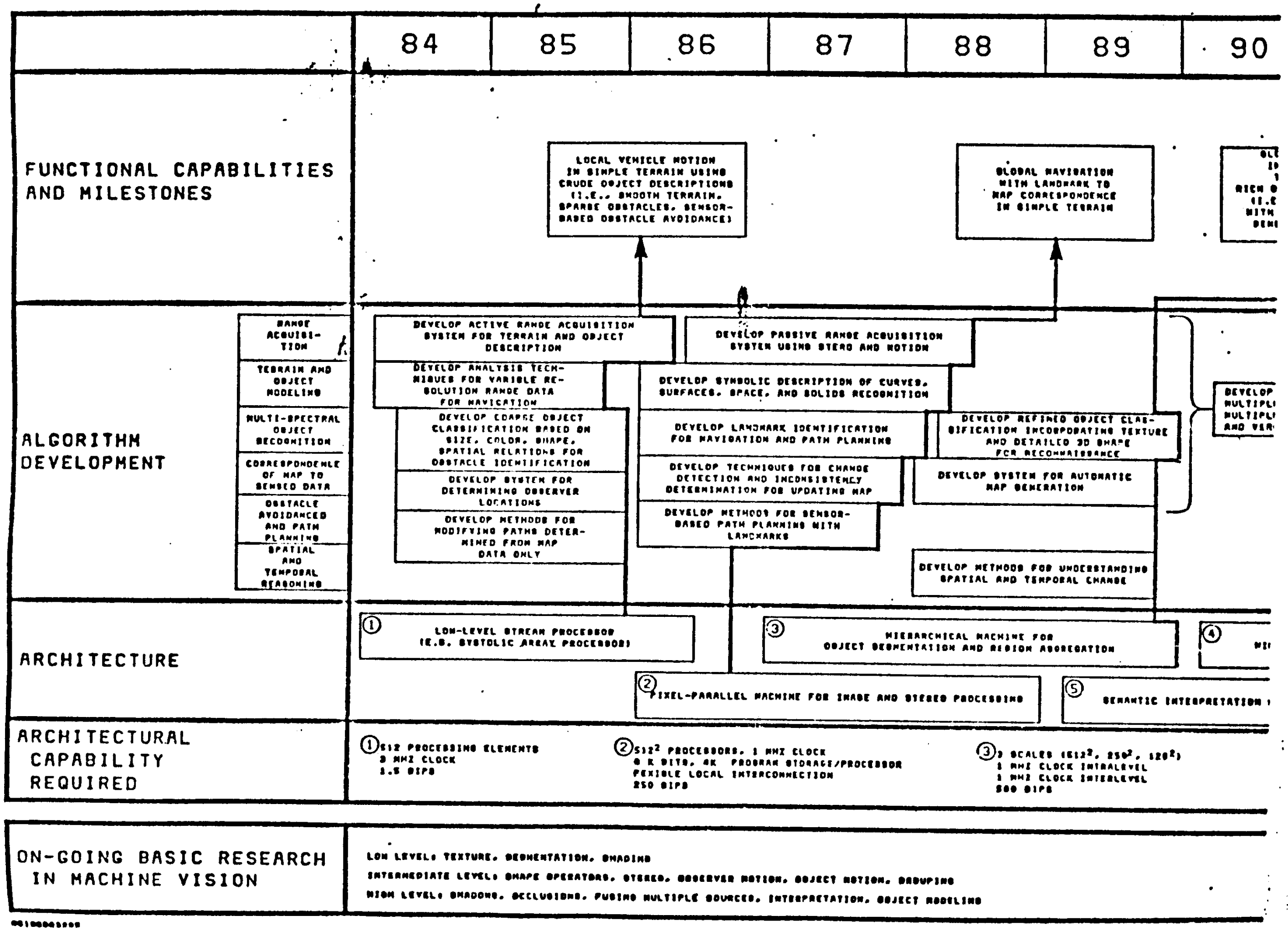


SPEECH SUBSYSTEMS

VERSION OF 10-18-83 II.1.2

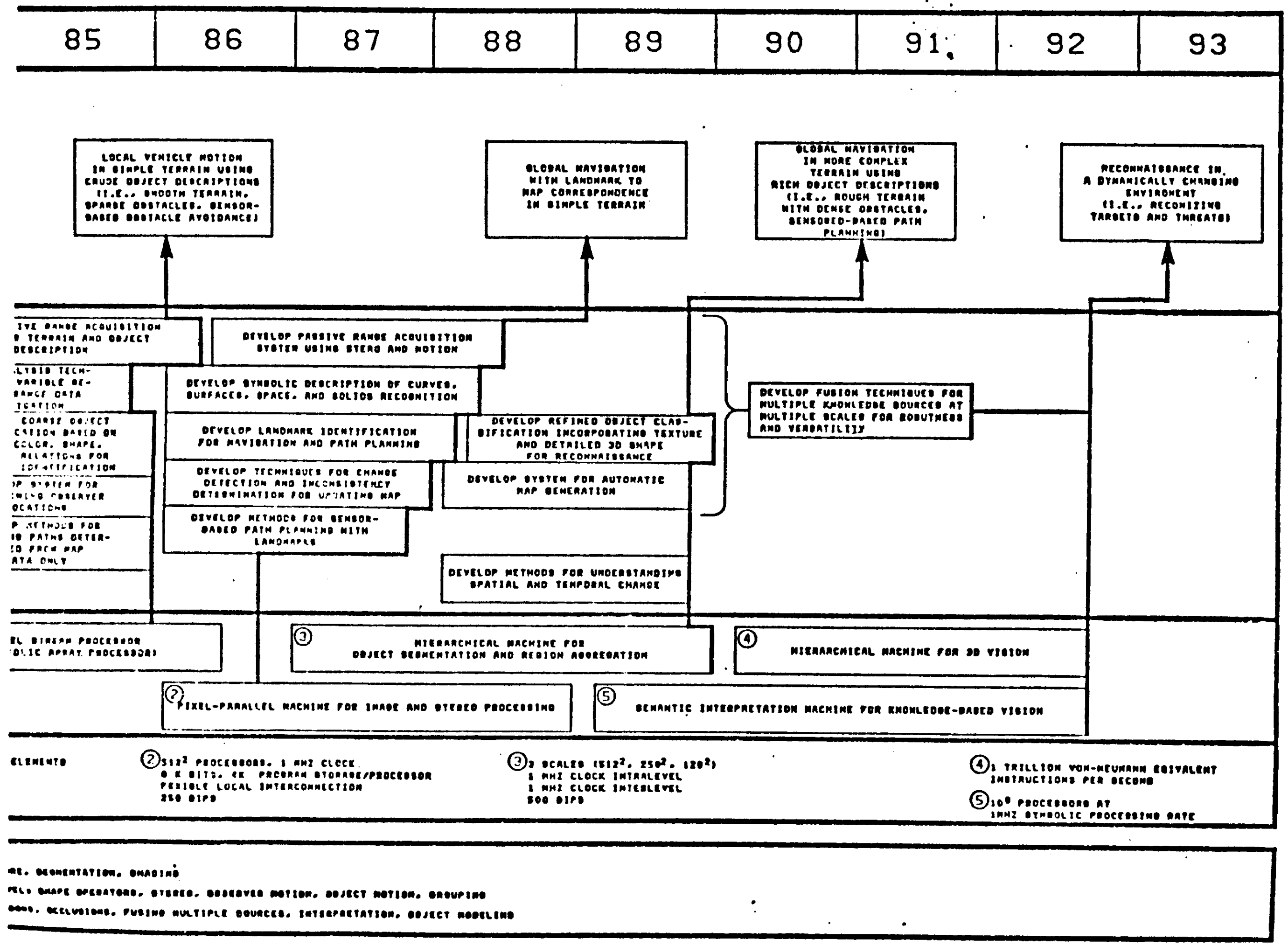


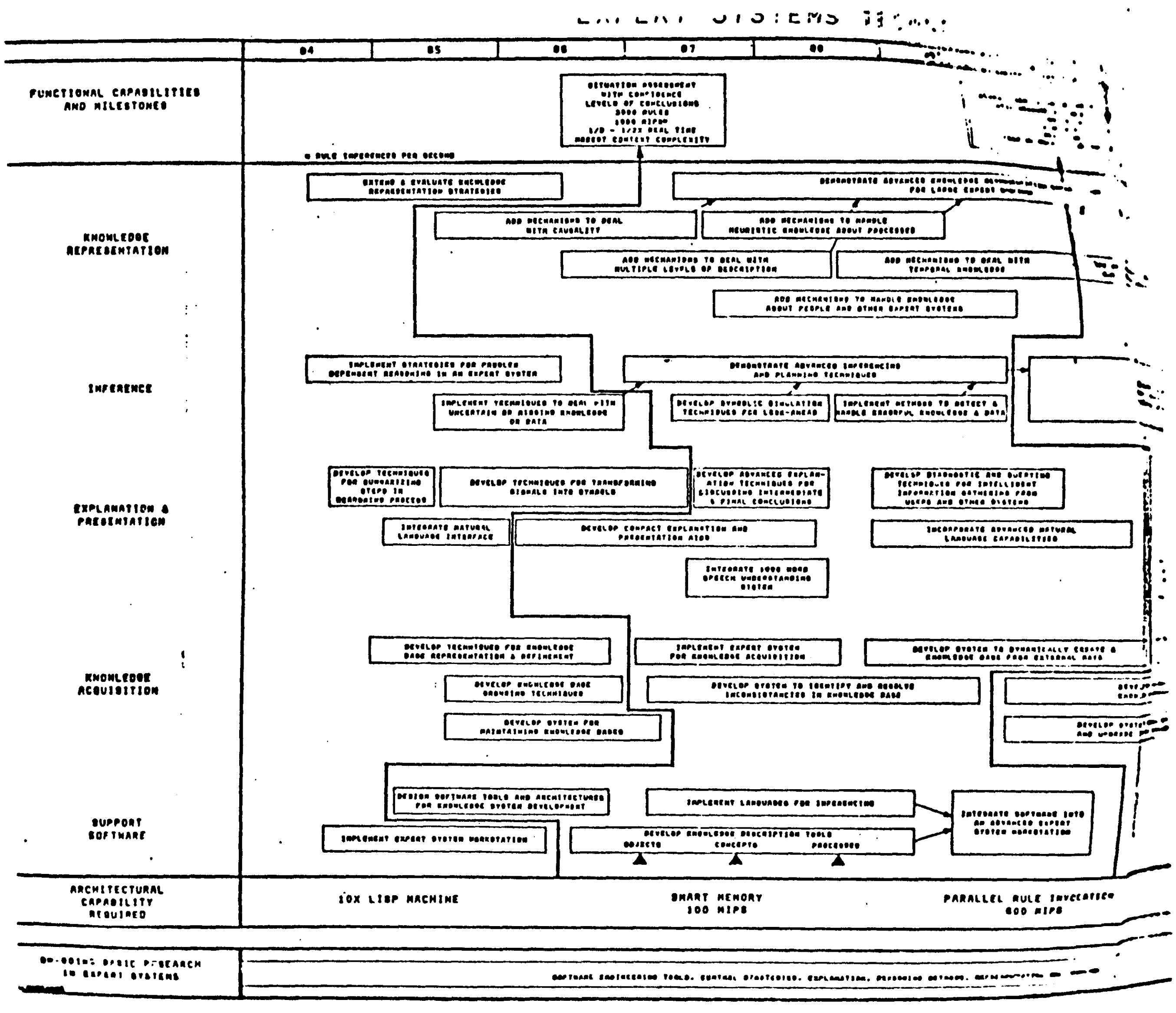
VISION SUBSYSTEMS

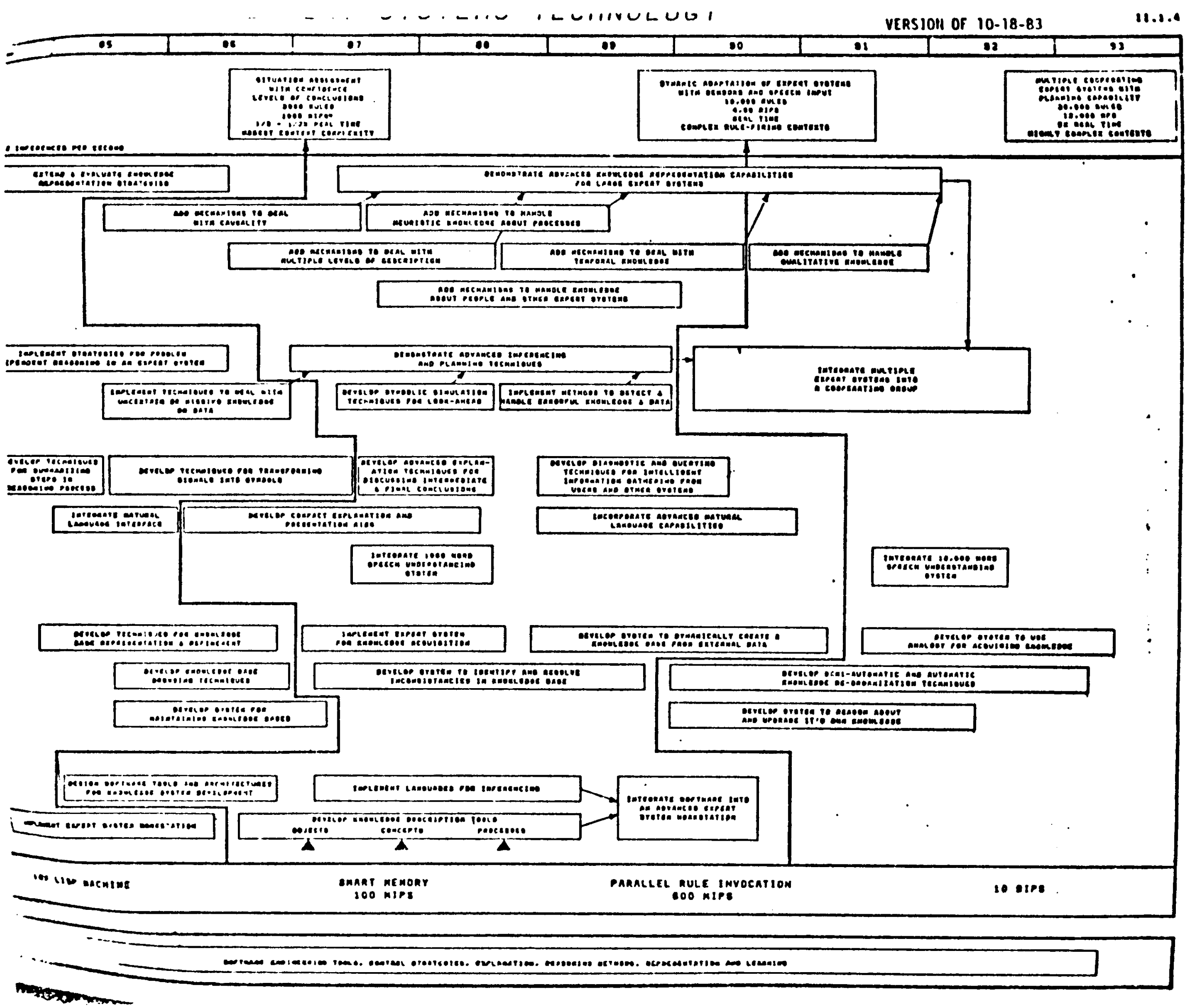


VISION SUBSYSTEMS

VERSION OF 10-18-83 II.1.1




84
85
86
87
88
FUNCTIONAL CAPABILITIES AND MILESTONES
SITUATION ASSESSMENT WITH CONFIDENCE LEVELS OF CONCLUSIONS 3000 RULES 1000 MIPS 1/5 - 1/2X REAL TIME MODEST CONTEXT COMPLEXITY
RULE INFERENCES PER SECOND
KNOWLEDGE REPRESENTATION
EXTEND & EVALUATE KNOWLEDGE REPRESENTATION STRATEGIES
DEMONSTRATE ADVANCED KNOWLEDGE ... FOR LARGE EXPERT ...
ADD MECHANISMS TO DEAL WITH CAUSALITY
ADD MECHANISMS TO HANDLE HEURISTIC KNOWLEDGE ABOUT PROCESSES
ADD MECHANISMS TO DEAL WITH MULTIPLE LEVELS OF DESCRIPTION
ADD MECHANISMS TO DEAL WITH TEMPORAL KNOWLEDGE
ADD MECHANISMS TO HANDLE KNOWLEDGE ABOUT PEOPLE AND OTHER EXPERT SYSTEMS
INFERENCE
IMPLEMENT STRATEGIES FOR PROBLEM DEPENDENT REASONING IN AN EXPERT SYSTEM
DEMONSTRATE ADVANCED INFERENCING AND PLANNING TECHNIQUES
IMPLEMENT TECHNIQUES TO DEAL WITH UNCERTAIN OR MISSING KNOWLEDGE OR DATA
DEVELOP SYMBOLIC SIMULATION TECHNIQUES FOR LOOK-AHEAD
IMPLEMENT METHODS TO DETECT & HANDLE ERRORFUL KNOWLEDGE & DATA
EXPLANATION & PRESENTATION
DEVELOP TECHNIQUES FOR SUMMARIZING STEPS IN REASONING PROCESS
DEVELOP TECHNIQUES FOR TRANSFORMING SIGNALS INTO SYMBOLS
DEVELOP ADVANCED EXPLANATION TECHNIQUES FOR DISCUSSING INTERMEDIATE & FINAL CONCLUSIONS
DEVELOP DIAGNOSTIC AND QUERYING TECHNIQUES FOR INTELLIGENT INFORMATION GATHERING FROM USERS AND OTHER SYSTEMS
INTEGRATE NATURAL LANGUAGE INTERFACE
DEVELOP COMPACT EXPLANATION AND PRESENTATION AIDS
INCORPORATE ADVANCED NATURAL LANGUAGE CAPABILITIES
INTEGRATE 1000 WORD SPEECH UNDERSTANDING SYSTEM
KNOWLEDGE ACQUISITION
DEVELOP TECHNIQUES FOR KNOWLEDGE BASE REPRESENTATION & REFINEMENT
IMPLEMENT EXPERT SYSTEM FOR KNOWLEDGE ACQUISITION
DEVELOP SYSTEM TO DYNAMICALLY CREATE & KNOWLEDGE BASE FROM EXTERNAL DATA
DEVELOP KNOWLEDGE BASE BROWSING TECHNIQUES
DEVELOP SYSTEM TO IDENTIFY AND RESOLVE INCONSISTENCIES IN KNOWLEDGE BASE
DEVELOP SYSTEM FOR MAINTAINING KNOWLEDGE BASES
SUPPORT SOFTWARE
DESIGN SOFTWARE TOOLS AND ARCHITECTURES FOR KNOWLEDGE SYSTEM DEVELOPMENT
IMPLEMENT LANGUAGES FOR INFERENCING
INTEGRATE SOFTWARE INTO AN ADVANCED EXPERT SYSTEM WORKSTATION
IMPLEMENT EXPERT SYSTEM WORKSTATION
DEVELOP KNOWLEDGE DESCRIPTION TOOLS
OBJECTS
CONCEPTS
PROCESSES
ARCHITECTURAL CAPABILITY REQUIRED
10X LISP MACHINE
SMART MEMORY 100 MIPS
PARALLEL RULE INVOCATION 600 MIPS
ON-GOING BASIC RESEARCH IN EXPERT SYSTEMS
SOFTWARE ENGINEERING TOOLS, CONTROL STRATEGIES, EXPLANATION, REASONING METHODS,


SYSTEMS TECHNOLOGY
VERSION OF 10-18-83
85
86
87
88
89
90
91
92
93
DEMONSTRATE ADVANCED KNOWLEDGE REPRESENTATION CAPABILITIES FOR LARGE EXPERT SYSTEMS
ADD MECHANISMS TO DEAL WITH CAUSALITY
ADD MECHANISMS TO HANDLE HEURISTIC KNOWLEDGE ABOUT PROCESSES
ADD MECHANISMS TO DEAL WITH MULTIPLE LEVELS OF DESCRIPTION
ADD MECHANISMS TO DEAL WITH TEMPORAL KNOWLEDGE
ADD MECHANISMS TO HANDLE QUALITATIVE KNOWLEDGE
ADD MECHANISMS TO HANDLE KNOWLEDGE ABOUT PEOPLE AND OTHER EXPERT SYSTEMS
DEMONSTRATE ADVANCED INFERENCING AND PLANNING TECHNIQUES
INTEGRATE MULTIPLE EXPERT SYSTEMS INTO A COOPERATING GROUP
IMPLEMENT LANGUAGES FOR INFERENCING
SMART MEMORY
100 MIPS
PARALLEL RULE INVOCATION
800 MIPS
10 BIPS

Chapter 8

References

---

2. Douglas R. Hofstadter, Godel, Escher, Bach, (New York: Basic Books, 1979) pp. 279-280.

3. Edward A. Feigenbaum and Pamela McCorduck, The Fifth Generation (Reading: Addison Wesley, 1983)

4. C. Northcote Parkinson, Parkinson's Law, (New York: Ballantine Books, 1975), p.15

10. Shannon, quoted in Pamela McCorduck, Machines Who Think (New York: W.H. Freeman and Company, 1979), p.101

11. M. Mathews and L. Rosler, "A Graphical Language For Computer Sounds," in Music by Computers, eds. H. von Foerster and J.W. Beauchamp (New York: John Wiley, 1969), p. 96

15. Richard H. Popkin and Avrum Stroll, Philosophy Made Simple New York: Doubleday & Company, 1956

17. Lady A.A. Lovelace, Notes Upon the Memoir "Sketch of the Analytic Engine Invented by Charles Babbage", by L.F. Menabrea (Geneva, 1842), reprinted in P. and E. Morrison, Charles Babbage and His Calculating Engines, pp.248-9, 284

21. Harvey J. Brightman, Lawrence L. Schkade, Dick Schoech. "Managing The Decision Sciences Technology: A Case For Ethical analysis?" Decision Sciences 12 (October 1981):4

22. See John T. Dunlop, ed., Automation: Implications For The Future. (New York: Vintage Books, 1962), Chapt. 6; John W. Kendrick and Ryuzo Sato, "Factor Prices, Productivity, and Economic Growth," American Economic Review 53 (December 1963):974-1003

23. Ibid., Chapt. 7

28. *Principia*, quoted in Pamela McCorduck, *Machines Who Think* (New York: W.H. Freeman and Company, 1979), p.34.

30. Milton Nahm, *Selections from Early Greek Philosophy*, (New York: F.S. Crofts, 1947)

31. George Berkeley, *Three dialogues*, ed. by M. Turbayne, (New York: Liberal Arts Press, 1954)

34. Interview with Gerhardt Freidrich, Digital Equipment Company, Boston, Massachusetts, 11 January 1984.

35. John von Neumann, *The Computer and the Brain*, (New Haven: Yale University Press, 1958) p.80

36. George Boole, *An Investigation of the Laws of Thought on Which Are Founded the Mathematical Theories of Logic and Probabilities*, 1854.

39. McCorduck, *Machines Who Think*, p.40

41. C.I. Lewis and C.H. Langford, *History of Symbolic Logic*, In *The World of Mathematics*, J.R. Newman, ed. (New York: Simon and Schuster 1956)

42. Ibid.

44. Howard DeLong, *A Profile of Mathematical Logic*, (Reading, Mass: Addison-Wesley, 1970)

45. Ernest Nagel and James R. Newman, *Godel's Proof*, (New York: New York University Press, 1958)

47. Russell and Whitehead, *Principia*

48. Andrew Hodges, *Alan Turing: The Enigma*, (New York: Simon and Schuster, 1983)

49. C. Reid, *Hilbert*, (George Allen & Unwin;.Springer Verlag, 1970) from Hodges, *Alan Turing*

50. Kurt Godel, *On Formally Undecidable Propositions*, (New York: Basic Books, 1962) The actual paper, in German, was: "Uber Formal Unentscheidbare Satze der **Principia Mathematica** und Verwandter Systeme, I." *Monatshefte fur Mathematik und Physik*, 38 (1931), pp.173-198.

51. C.E. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," *Trans. Amer. I.E.E.* 57 (1938)

53. Hodges, *Alan Turing*, p.91

54. McCorduck, Machines Who Think, p. 41

55. Hodges, Alan Turing, p. 295

56. Alan M. Turing, "On Computable Numbers, with an Application to the **Entscheidungsproblem.**" Proc. Lond. Math. Soc. 2, 42 (1937)

60. Alan M. Turing, "Computing Machinery and Intelligence," Mind Vol. LIX, No. 236 (1950)

61. Hodges, Alan Turing, p.251

62. Norman Weiner, Cybernetics, 2nd ed. (Cambridge: MIT Press, 1961)

66. quoted in Hodges, Alan Turing, p.417

68. Hodges, Alan Turing, p.112

70. Hofstadter, Godel, Escher, Bach, p.429

72. Hubert Dreyfus, What Computers Can't Do; A Critique of Artificial Reason, (New York: Harper and Row, 1972)

73. Stephen Jay Gould, Ever Since Darwin (New York: W.W. Norton & Company, 1977)

75. Laurance Urdang, ed., The Random House Dictionary of the English Language, College Edition (New York: Random House, 1968)

78. Burke, James. Connections. (Boston: Little, Brown and Company, 1978), p. 106

79. Ibid., p. 108

91. Interview with Al Stevens, Bolt, Beranek, and Newman, Cambridge, Massachusetts, 6 May, 1983

96. C.E. Shannon, "A Mathematical Theory of Communications," Bell System Tech. Journal (July 1948 and October 1948).

97. T. Berger, Rate Distortion Theory, (Englewood Cliffs: Prentice Hall, 1971)

98. R.W. Hamming, "Error Detecting and Error Correcting Codes," Bell System Tech. Journal 29 (1956)

99. Van Nostrand's Scientific Encyclopedia, Fifth ed., s.v. "Assembler."

112. Avron Barr and Edward A. Feigenbuam, The Handbook of Artificial Intelligence, Vol. II, (Stanford: HeurisTech Press, 1982

113. A. Barr, "Meta-Knowledge and Cognition", International Joint Conferences on AI 6 (1979) pp. 31-33

114. Hofstadter, Godel, Escher, Bach, p. 602.

118. Paul Kinnucan, "Computers That Think Like Experts," High Technology 4 (January 1984) pp. 30-42

119. Samuel Holtzman, "Artificial Intelligence; Basic Concepts, and Introduction to Expert Systems" (Presentation, Department of Engineering-Economic Systems; Stanford University)

120. Frederick Hayes-Roth, Donald A. Waterman, and Douglas B. Lenat eds., Building Expert Systems (Reading: Addison-Wesley, 1983)

121. Carl Sagan, Broca's Brain, (New York: Random House, 1974)

122. Patrick Winston, Artificial Intelligence, (Reading: Addison-Wesley, 1979) , pg. 1.

123. Samuel Holtzman, "Artificial Intelligence, Basic Concepts and an Introduction to Expert Systems"

124. S.C. Kleene, Mathematical Logic, (New York: John Wiley and Sons, 1967); P. Suppes, Introduction to Logic, (Princeton: D. Van Nostrand Reinhold, 1957)

125. Paul Y. Gloess, Understanding Artificial Intelligence, (Sherman Oaks: Alfred Publishing, 1981)

127. Hofstadter, Godel, Escher, Bach, p.644.

128. Paul Kinnucan, "Computers That Think Like Experts" High Technology (January 1984) p.33

129. R. Davis, B.G. Buchanan, and E. Shortliffe, "Production rules as a representation for a knowledge-based consultation program." Artificial Intelligence 8 (1977) :15-45

130. E.H. Shortliffe, Computer bassed medical consultation: MYCIN. (New York: American Elsevier, 1976)

132. R.O. Duda, P.E. Hart and N.J. Nilsson, "Subjective Bayesian methods for rule-based inference systems." In Proceedings of the AFIPS 1976 National Computer Conference, 45 (1976) pp. 1075-1082

133. Preceding examples adapted from Hayes-Roth, et, al, Building Expert Systems, pp. 94-97

135. J. Doyle, "A model for deliberation, action, and introspection" (Ph.D dissertation, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980)

136. B. Hayes-Roth and F. Hayes-Roth, Cognitive Processes in Planning (Santa Monica: RAND Corporation, [1978],

137. M. Stefik, "Planning With Constraints" (PhD. Dissertation, Stanford University, 1980)

138. Herbert Simon, The New Science of Management Decision, (Englewood Cliffs: Prentice-Hall, 1965)

139. Chuck Williams, "ART The Advanced Reasoning Tool, A Conceptual Overview" explanatory paper Inference Corporation, Los Angelos

141. J. McCarthy, "History of LISP" SIGPLAN Notices 13 (1978) pp. 217-223

144. Steven Shwartz, "Natural Language Processing in the Commercial World," in Artificial Intelligence Application for Business, ed. Walter Reitman (Norwood: Ablex Publishing, 1984), p. 235.

145. Madeline Bates and Robert Bobrow, "Natural Language Interfaces: Whats Here, What's Coming, and Who Needs It," in Artificial Intelligence Application for Business, ed. Walter Reitman (Norwood: Ablex Publishing, 1984), p. 179

146. W.A. Woods, "A personal view of natural language understanding," SIGART Newsletter 61 (February 1977) pp. 17-20

148. Shwartz, Ibid.; Woods, Ibid.

149. Woods, Ibid.

153. Madeline Bates and Robert Bobrow, "Natural Language Interfaces: Whats Here, What's Coming, and Who Needs It," in Artificial Intelligence Application for Business, ed. Walter Reitman (Norwood: Ablex Publishing, 1984), pp. 186-187.

154. Bonnie Webber and Tim Finn, "In Response: Next Steps in Natural Language Interaction" in Artificial Intelligence Application for Business, ed. Walter Reitman (Norwood: Ablex Publishing, 1984)

158. U.S., Department of Defense, Strategic Computing, New-Generation Computing Technology: A Strategic Plan for its Development and Application to Critical Problems in Defense, p.7.

159. Samuel Holtzman, "Artificial Intelligence, Basic Concepts and an

Introduction to Expert Systems"

161. R. Davis "Expert systems: Where are we? And Where Do We Go From Here?" MIT Artificial Laboratory, AI Memo No. 665, June, 1982

163. Drawn from; Harry Pople, "Knowledge Based Expert Systems: The Buy or Build Decision." in Artificial Intelligence Application for Business, ed. Walter Reitman (Norwood: Ablex Publishing, 1984), p. 27 and R. Hayes-Roth, et al. Capt. 5

170. Hayes-Roth, et al; Building Expert Systems