

UC Berkeley

UC Berkeley Previously Published Works

Title

Building the Second Mind, 1961-1980: From the Ascendancy of ARPA to the Advent of Commercial Expert Systems

Permalink

<https://escholarship.org/uc/item/82h464gg>

Author

Skinner, Rebecca Elizabeth

Publication Date

2012-09-03

**Building the Second Mind, 1961-1980: From the
Ascendancy of ARPA to the Advent of Commercial Expert
Systems**

Rebecca E. Skinner

Copyright 2013 by Rebecca E. Skinner
ISBN 978-0-9894543-2-2

Forward

Part I. Introduction

Preface

Chapter 1. Introduction: The Status Quo of AI in 1961

Part II. Twin Bolts of Lightning Alter the Landscape

**Chapter 2. The Revolutionary Impact of the Integrated
Circuit**

**Chapter 3. The Revitalization of the Advanced Research
Projects Agency and the Foundation of the IPTO**

**Chapter 4. Hardware, Systems and Applications in the
1960s**

Part III. The Belle Epoque of the 1960s

Chapter 5. MIT: Work in AI in the Early and Mid-1960s

**Chapter 6. CMU: From the General Problem Solver to the
Physical Symbol System and Production Systems**

Chapter 7. Stanford University and SRI

Part IV. The Challenges of 1970

**Chapter 8. The Mansfield Amendment, "The Heilmeier Era",
and the Crisis in Research Funding**

**Chapter 9. The AI Culture Wars: the War Inside AI and
Academia**

Chapter 10. The AI Culture Wars: Popular Culture

Part V. Big Ideas and Hardware Improvements in the 1970s

**Chapter 11. AI at MIT in the 1970s: The Semantic Fallout
of NLR and Vision**

**Chapter 12. Hardware, Software, and Applications in the
1970s**

Chapter 13. Big Ideas in the 1970s

Chapter 14. Conclusion: the Status Quo in 1980

Chapter 15. Acknowledgements

Chapter 16. List of Abbreviations

Chapter 17. Bibliography
Chapter 18. Endnotes

Forward to the Beta Edition

This book continues the story initiated in Building the Second Mind: 1956 and the Origins of Artificial Intelligence Computing. Building the Second Mind, 1961-1980: From the Establishment of ARPA-IPTO to the Advent of Commercial Expert Systems follows the history, through to the fortunate phase of the second decade of AI computing. How quickly we forget the time during which the concept of AI itself was considered a strange and impossible effort ! The successes and rapid progress of AI software programs, with related work in adjacent fields such as robotics and complementary fields such as hardware, thus are our theme and meat of our narrative.

The Building the Second Mind venture traces back to the author's Ph.D. dissertation, which studied the commercialization of AI during the 1980s. Curiously, portions from the author's original Ph.D. dissertation itself, concerning the abortive business enterprises, working in fits and starts, devoted to early commercialization of expert systems, now make up several chapters of the third volume.

Most of AI is found in ongoing laboratory research, published in dense academic journal articles and in conference proceedings. Synoptic and simplifying works, such as this book, are apt to appear superficial to the researchers who write such papers, yet overly complex to the man on the street. The author tries to avoid both of these ills. Popularizers are not well-liked in academia or the world of research. However, with the exception of those who write the journal articles, everyone is someone's popularizer. They are a necessary part of the intellectual ecosystem, since sci-fi novels and movies alone can't do the work of representing research. This work is neither highly technical nor comprehensive: my interests are skewed in favor of knowledge representation. Building the Second Mind draws heavily upon both academic and journalistic nonfiction

sources regarding AI. Much work in AI is necessarily specialist, but the value of synoptic works should not be underestimated.

New technological advances allow alterations of the traditional publishing conventions, leaving social and professional convention to adjust. Paper publishing schedules and the pace of review have historically slowed publication and precluded revisions. This author instead asks for editorial input from the Cloud for this Beta edition. This will result in far more reviews from a larger pool of knowledgeable people. This will make it far easier for this text to include hyperlinks, refs to oral histories and museum and scholarly websites, as well as corrections and editorial improvements. I will take comments for several months and then release a revised ePub edition later in 2012.

Preface

Building the Second Mind, 1961-1980: From the Establishment of ARPA-IPTO to the Advent of Commercial Expert Systems tells the story of the development of AI, the field of research that sought to get computers to do things that would be considered intelligent if a person did them, during the 1960s and 1970s. (The definition is commonly attributed to Marvin Minsky). The book takes its leave at the conclusion of the '70s, shortly before both expert systems and personal computing were commercially introduced in their awkward early forms. In the late 1950s, the field had been founded and began to undertake extremely rudimentary logic and problem-solving programs. In the 1960s, the immense growth in funding given to the field of computing, the development of integrated circuits for mainframe computing, and the increasing numbers of people in AI and in computer science in general, all tremendously advanced the field. This is evidenced in more complex problem-solving, development of an understanding of the field of knowledge representation; appearance of fields such as graphic representation and problem-solving for more knowledge-intensive fields. Finally, the early integrated circuits of the 1960s gave way to microprocessors and microcomputers in the early 1970s. This heralded a near future in which computers would be increasingly fast and ubiquitous. In a clear virtuous cycle, the enhanced

cheapness of processing power and storage would encourage the proliferation of applications.

This work takes up where Building the Second Mind: 1956 and the Origins of Artificial Intelligence Computing ended. That book studied this field from the distant prehistory of abstract thinking, through its early formation as a field of study in the mid-1950s, and the several years following that turning point. Watching the advances of the 1960s and 1970s offers the satisfying vindication of the early efforts of AI's founders- John McCarthy, Marvin Minsky, Allen Newell, and Herbert Simon.

Enthralled with Technology

“ O BraveNewWorld thathassuchpeoplein it !” AldousHuxley, BraveNewWorld. 1932.

The early 1960s was an expansive, ebullient era. The NASA mission to land a man on the Moon, initiated in 1961 and achieved in 1969, was one of the most memorable events of both the Cold War and scientific and technical history. During the Cuban Missile Crisis of 1962-1963, the United States confronted Cuba, and by proxy the USSR, demanding that the Soviets refrained from stationing missiles in the Caribbean. The peaceful resolution of this showdown proved that the USA at least held its own, and could keep the Soviets from maintaining a military presence in the Western Hemisphere. The conquest of diseases through vaccines continued. The cultural fascination with the products of technological modernity continued: new materials in the form of novel fabrics, building materials, and jarring and atonal music and art, ensured that cultural enlightenment included an element of discomfort and shock. High Modernist architecture provided exposed metal and glass, with enough sharp edges and right angles to fill a geometry textbook. Freeways plowed through cities: an elevated highway wrecked the San Francisco waterfront until an earthquake took it down, and a highway nearly was run through the dense streets of Greenwich Village in New York City. The sleek and metallic and manufactured and aerodynamic were lauded; the quaint and candlelit and rustic had no place.

Modernism in the form of the wholesale glorification of the new, including faith in AI and the progress offered by computation, would see profound challenges. The central cultural shocks which began to break Modernism apart had not taken place yet. But they were beginning to appear: Jane Jacobs' influential masterwork The Death and life of great American cities appeared in 1961 (1), and Silent Spring by Rachel Carson was published in 1963 (2). Jacobs helped to bring about the new urban movement of densely populated, village-like urban centers, and Silent Spring decried environmental pollution. Both helped to bring caveats to technological growth, but the early 1960s saw only the very beginning of doubts as to its munificence. Likewise, the ardent participation of both parties in the Cold War had not yet begun to show fatigue.

Building the Second Mind will be an advocate of computer technology as an augmentation technology for human abilities and as a research technology for cognitive science. General belief in- if not blind adherence to- scientific progress has been an occupational requirement for writing this work. However, the caveats to technological progress issued by such challenges to Modernism are significant, and will be considered later in this book.

The Context of AI's Progress

Artificial Intelligence was one of the great triumphs of Postwar technological optimism, and this time period was one of its sweet spots. During the 1960s and 1970s, the field developed tremendously. It started to devise ways to represent visual data by computation, as in the 1966 MIT Summer Vision project; began to examine the mysteries of creativity (in the work of Herbert Simon and his students); tore apart sentences for their meaning (Schank conceptual primitives, Bobrow calculus problem parser); and developed means to represent high-level scientific problem-solving and classification knowledge (Buchanan and Feigenbaum's Dendral program), among other things.

AI's earlier history, in the years of the postwar research and development frenzy, had been more precarious. After the Second World War, the idea of making machines think had seemed strange and disturbed. Beyond this, it had often been perceived as morally

bad, and the whole concept sinister. The appearance of arguments in its favor, or at least those urging measured equivocation, was a welcome salve. But by the late 1960s, the situation was immensely different. The shoe was on the other foot, and the field had achieved success with chess-playing programs, with the Logic Theorist, with calculus-solving programs, with early robotic arms, and most recently with the Dendral program, which solved diagnoses in internal medicine. With the appearance of the Advanced Research Projects Agency (ARPA, known as Darpa starting in 1972), AI had attained research centers, tenure-track professors, and influence in large-scale, capital-intensive timesharing projects. Throughout the 1970s the tasks which the field was capable of doing proceeded to be come far more erudite, as better forms of abstraction and embodiment of knowledge were created. As Edward Feigenbaum, a Stanford researcher who will be important to this book said, the field had a great deal of low fruit, easily attained, to pick.

This book tells this story.

2. Themes

The thematic concerns of this work should not overshadow its historical nature. We recognize that there is a voluminous and enthusiastic body of knowledge regarding the history and sociology of technology and science. Our concern with contribution to this body of knowledge is first and foremost expository rather than theoretical. We will revisit the themes of the last book. In other cases they are revised because of the change in the lessons of history.

First, the successive successes of AI are salient in their most unalloyed form in this time period. During the 1950s all advances were painstaking and even the most simple thing took astonishing effort. During the 1960s, and into the 1970s, AI successively knocked down the list of things that had seemed that it would not be able to do. Herbert Simon's 1963 scholarship began to analyze creativity in a systematic fashion. Programs involving schematized worlds, or "toy" domains, were devised. More difficult domains, such as the systematic but complex field of mass spectrometry,

were and AI programs analyzed spectrometric data. This was accomplished by a program known technically as a production systems, which would later be developed as “expert systems” in the 1980s. Vision, robotics, computer chess, and knowledge representation all underwent new and fundamental changes.

Second, the theme of the successful implementation of technological innovation by governmental programs- war both hot and cold, is repeated (3). The author said it in the beginning of *Building the Second Mind: 1956 and the Origins of AI Computing*, and shall say it again here. The role of the military in the initial birth and later development of the computer and its ancillary technologies cannot be erased, eroded, or diminished. As we saw in the first book, this was true of the earliest development of the digital computer during the Second World War, and of its progress along a broad variety of fronts during the postwar decades. The advances which may be directly attributed to the war effort- the Cold War effort, that is- include relay memory followed by magnetic core memory; storage in the form of punched cards and magnetic tape; input-output in the form of followed by teletype operations, and finally the CRT screen with a keyboard. The crucial role of government policy in expressly and successfully encouraging computer development is a well-articulated scholarly theme (4). Our study is simply a further corroboration.

Third, a well-tuned organizational culture was also essential in the development of the computer in every aspect, along the course of the entire chronology. ARPA allowed and encouraged creative people to work on their own terms, at a multiplicity of institutions- the Rand Corporation, SRI, and the laboratories at each of the universities, among others. The same results would not have taken place without the ARPA bureaucracy and the institutional freedom which it encouraged.

A **fourth** theme is that of the paradigm transition in scientific and popular conception of computers and machinery entailed by AI. The essay of AI itself envisions an active intelligence apart from humans. Getting this new gestalt in place has required the incipience of acceptance of the computer, and a rethinking of the capabilities of human tools. Belief in the computational metaphor

for the human mind, and belief in the possibility of computing as qualitative as well as only numerical and quantitative information processing, were established among a very small cohort by the time discussed in the beginning of this book. The corner had already been turned in these erudite circles, but the decades of the 1960s and 1970s saw the vast widening of the software to prove that programs could emulate cogitation, and the hardware which could carry this out. The proof was in the software.

Fifth and finally, our study is concerned with the profound contributions of AI beyond the area of its designated efforts- to build tools that do 'intelligent' things and to emulate human intelligence. Much of the achievement of the field during this time consisted of AI researchers' development of computer languages and operating systems more generic to research and consumer computing (the Multics program, which helped lead to C++ and Unix); more interactive input-output in the form of input through a keyboard, with a monitor one looks at; and useful computing tools such as newswire programs, editing conventions, and the concept of software agents for a wide range of tasks. Put in even simpler terms: AI and its wider circles was absolutely central to the development of the personal computer and the popular Internet.

Outline of the Chapters

The history of AI through 1956, and in the several years afterwards, naturally structures itself with the culmination of field's foundation. The history of AI during the 1960s and 1970s naturally structures itself as well. The onslaught of monies given to AI and other computing disciplines, following the 1961 selection of ARPA to be the pre-eminent research agency for funding computer science, was a game-changing windfall. It could be compared to a television or movie sight gag in which the door is opened and a flood ensues. Thus the narrative begins with a bang, imposed by history, as the increased effort in the United States' space program meant increased funding for AI as well.

We commence by presenting the framing facts of the integrated circuit (IC), which has allowed computers that offer bigger, better, faster, and more computing, and ARPA, which footed the tab for its

development. These were invented and founded in 1959 and 1958 respectively, but did not become prominent until their country needed them, so to speak, because of the NASA Apollo mission. The 1962 foundation of ARPA's Information Processing Techniques Office, and its immeasurable contributions in funding much AI-related research during these two decades, are discussed as it specifically relates to AI's agenda (**Chapters Two and Three**). The wide dispersal of improvements to input-output, memory, and storage are discussed in **Chapter Four**.

Part III. The Belle Epoque of the 1960s

The sine qua non of ARPA's contribution to AI was its free rein to researchers. AI flourished in response to opportunity unfettered. We refer to this decade as the Belle Epoque, in reference to the salubrious climate for art and literature prior to the First World War. In Chapter Five we discuss the more general task of institution-building that nurtured AI.

At the Carnegie Institute of Technology (now Carnegie Mellon University), Newell and Simon expanded upon the schematic and simple problems they had initially taken on with the General Problem Solver program, and began to work on the production system protocol for solving toy problems. These programs, which were seminal for cognitive science, were generally not concerned with domain knowledge, but with the nature of accretion of knowledge in small, closely defined 'toy' domains such as games and SAT-type problems (**Chapter Six**). Today computer programs can solve hard problems; at that time solving tiny problems was an enormous achievement. Closely influenced by Newell and Simon (and thus included in the CMU chapter), Bruce Buchanan and Simon's protégée Edward Feigenbaum at Stanford, initiated the Dendral project, AI's first significant success in solving a problem in a difficult scientific domain.

Chapters Five and Seven address AI at MIT and Stanford respectively. Embodiment of intelligence through vision and robotics proceeded with work at the MIT AI Lab, and at the Stanford AI Lab ('SAIL'), established by John McCarthy in the early 1960s. Concepts such as the representation of graphic imagery by pixels were first

established, apparently at MIT; hacking continued round the clock, and led to numerous programming and applications innovations. MIT and Stanford in particular were hotbeds of primordial applications, at the time used by dozens of people. These include online news readers; editing and e-publishing; computer music; mathematical problem solvers; drawing and graphing programs; and the infamous finger command to locate users online. Today such things appear as ubiquitous “apps” used by billions of people. McCarthy was one of the main inventors of timesharing, which was commercialized in the late 1960s. The progressive implementation of timesharing both drove and responded to demand for computing resources, resulting in a series of ARPA projects to implement networked remote terminals and central servers for increasing numbers of users.

Ebullience typically gives way to disillusionment, just as sunshine gives way to fog. Technological novelty often is demeaned when capricious cultural trends vote against it, in the process taking advances for granted. Chapters Eight, Nine, and Ten show how the tide of national research politics, as well as elite university and popular culture, began to question AI during the early and middle 1970s.

The ignominious ending of The Vietnam War cast a shadow on the larger imperial defense purposes of the United States. Whatever the merits of the American imperial mission during the Cold War, the struggle for military supremacy against the Soviets was good for AI. The defeat (or whatever euphemisms were applied) in Southeast Asia led to the questioning of defense funding and, this doubt in turn tightened the screws on AI research monies. George Heilmeier, a brilliant and controversial tactician who fought the battle to keep the field both research-oriented and capable of applications, worked effectively with AI’s research leaders to mediate between the Pentagon’s brass and the less tractable world of endogenously-driven research **(Chapter Eight)**.

Curiously, both AI and the world outside it doubted the field during an immensely fruitful time period. AI began to criticize itself, both as regards its own fundamental postulates, its emphasis of cogitation, and its (putative) leanings toward logical positivism

(Chapter Nine). Any familiarity with AI's founders indicates that none of them believed that intelligence was solely embodied in IQ and SAT test type problems, much less in Lisp or IPL code. The first AI program run- the Logic Theorist implemented by Newell and Simon at Carnegie Tech at the end of 1955- was carried out by grad students with cards to indicate data. AI was clearly going to work its way up almost from the beginning of computing itself. However, much of the straw man critique, even from inside AI seemed to buy into such a simplifying scenario.

For separate and independent reasons, the outside world of highbrow and pop culture was especially critical of synthetic intelligence as well. During the late 1960s and into the murky middle of the 1970s, a pernicious anti-technological philosophy was quite well-respected. Perhaps evidencing the fell hand of Martin Heidegger, pop phenomenology called into question rational cognition (as opposed to raw experience), and scientific progress and its works. AI is nothing if not implicitly and explicitly faithful in progress, and often the field took it square in the jaw in pop culture jabs against scientific progress. Remaining unruffled in the face of criticism is one trait of the mature and self-confident: AI certainly met this challenge. This led to ironies such as Marvin Minsky consulting to the completion of 2001: A Space Odyssey, a movie in which a computer program achieves sentience and self-conscious purpose, murders a human, and subsequently is dismantled and thus euthanized by humans **(Chapter Ten)**. The usually inarticulate anti-technological period is perhaps best exemplified in the infamous and sinister words: "I'm sorry Dave. I'm afraid I can't do that".

In Chapter Seven we indicated that vision and robotic embodiment were long-running and enormously consequential traditions established at MIT as early as 1960. Later in the 1960s and in the 1970s, vision took on the issue of semantics in a big project which encompassed a number of dissertations. This was the Microworlds project, which consisted of a semantics encompassing what was in essence a number of packing boxes and other geometrical figures on a floor. Vision, it seems, spawns the question of semantics of understanding what is being viewed. SHRDLU, one of the family of dissertations, was undertaken by Terry Winograd. SHRDLU's blocks

world resided on a TV screen. A [virtual] robotic arm could be ordered to move the blocks by commands typed in simple natural language (**Chapter Eleven**).

Other aspects of computing- processing power, memory, storage, monitor quality, and input-output quality grew enormously during the 1960s and 1970s, part of a growing commercial market at first dominated by enormous corporate clients but increasingly made up of countless smaller enterprises as well. This made it easier to develop applications; in a virtuous cycle from which all of us benefit today, this increased the ease and allure of computer research (**Chapter Twelve**).

The generally parallel concepts of scripts, frames, and production systems succeeded in chunking large problems in rich domains, rather than the toy domains in which AI had to initiate its research (**Chapter Thirteen**). Despite the differences in background in these conceptual frameworks, all of them attempted to write programs that could conceptualize knowledge representation for much larger cognitive acts. The concept of Scripts (or 'MOPS'), proposed by Yale professor Roger Schank, proffered the basic structure of knowledge consisted of repeated scenarios, in context of which one generally knows what to expect. Frames, conceived by Minsky, suggests that knowledge is represented in a structure with attributes for descriptive data, rules, default assumptions and other fundamental features. Production systems, which had initially been suggested by logician Emil Post in 1943, were formalized by Newell and Simon in the late 1960s. These essentially consist of a rule base for the means to respond to given information such that if a given A appears, B will be the response. Production systems are generally considered to be the basis for expert or knowledge base systems.

These large structural approaches to sorting through bodies of knowledge differed in their design but not in their comparable sophistication. As the 1970s ended, hardware was increasingly competent to manage such systematic approaches, and a number of the programs ran so well that their creators were beginning to try to generalize their lessons to commercial applications. Less commonly appreciated is the fact that a much larger boom in commercial applications of AI was about to appear in the form of

the personal computing revolution. The primordial features of the pc, including a keyboard for input, input through a screen, passwords, email, demon agents for software tasks, among other things, had been developed at AI's laboratories or in its immediate social and professional vicinity. Clearly, it was a good two decades.

Chapter 1. Introduction: The Brave New World of the 1960s

Our Story thus Far

The earliest years of AI, in the mid-1950s, witnessed the difficult implementation of computer programs to carry out 'toy problems'. These were simple, schematic, contained no idiosyncratic information, and could be summarized in tiny sets of rules. By the early 1960s, the problems were slightly more discursive- games, IQ-test type problems, and mathematics problems. But computer time was difficult to procure, even at major universities, and so all research still took place in a small search space. This does not mean that the future as perceived was not bright.

In 1961, as this story begins, worry about the future of computing, technology in general, and the glory of progress are far from anyone's mind. On the contrary: 1961, as we will see, was a time of bold initiatives. In this chapter, we summarize the larger scope of the state of the art in AI, and the expansive and technology-enthralled cultural environment. This chapter defines the starting point of BTSM, clarifying the status quo of AI circa 1961. This is a natural point of departure because the early history of the field as such breaks apart from the intense development of the 1961 at this time. In this book's predecessor, we studied the early history of AI in its formation.

As we saw, AI is an apparently new field that is actually ancient. The dream and then the concrete effort to mechanize aspects of human intelligence through machines is as old as humanity. In AI, Rationalist philosophy, which essentially sees the mind as a mechanism, and Empiricism, which focuses on the reality of felt experience, are brought together and reconciled with the appeal of artifacts and automata in Western technology.

Mechanical automata are the most obvious efforts at AI, and are the most historically prevalent. Yet these were not the most important developments early in the 20th century. Instead, the statements of concern to us at that time were more abstract. Formal logic, greatly and steadily amplified throughout the 19th century and the first half of the Twentieth, was used as a means to express relations between abstract logical symbols. During the 1930s, the predicate calculus was introduced. This was not mathematical calculus strictly speaking, but instead a formal language made up of logical statements which could define an object and then describe it.

Information theory- the first concept of which aligned formal logical statements with the on and off code of electrical signals- wedded electrical computing to formal logic in the late 1930s. Claude Shannon and Warren Weaver invented information theory in the years before the start of the Second World War. This was the essential first step after which computer languages- formal languages put into the form of electronic codes- could be developed. The electrical engineering conventions associated with information theory improved the transmission quality and verisimilitude of electrically transmitted data. The general-purpose digital computer was developed during the Second World War, then immensely improved after that war concluded.

During the early 1950s, logicians and psychologists began to wonder to what extent computer programs could be written that reiterated the human activities of problem-solving, or 'thinking'. The first AI program, the Theorem Prover, was first demonstrated at the end of 1955 by Allen Newell, Herbert Simon and engineer Cliff Shaw- of whom much more later. Mathematician John McCarthy, working with Marvin Minsky, Claude Shannon and others, convened the Dartmouth Summer Conference, held at that university during 1956. This meeting, and the subsequent IRE conference at Cambridge in September 1956, named the field and drew together its major founders in a loose and contentious group.

Thus the field of Artificial Intelligence coalesced in 1956. Yet AI became institutionally entrenched at MIT, and at Stanford during 1961 and 1962. As we will see later, the Stanford Research Institute

already had begun work in computing and many other fields of research. Joining the long-running work of Newell and Simon at the Carnegie Institute of Technology (renamed Carnegie-Mellon University in 1968), this triumvirate of major centers of research allowed for the time and resources that were necessary to get things done. In this book we study the long course of its development as a field for the next twenty years. The expansionary phase of these twenty years would be underwritten by the Defense Department, and productive in its multifold efforts.

The Status Quo of AI, and Computing, in 1961

Thus in the early few years, the field carried out problems appropriate to the initial exposition of problem solving. We will see by looking at the state of the art- with reference to a major article summarizing this at the time- that the reach exceeded the grasp, and that the conceptual nature of the field was far more than that.

The first problems were simple and schematic “toy” problems, with solutions that could nearly be anticipated. They illustrated pure problem-solving, but were thin on the issue of the domain. For two reasons, these nearly had to be the first problems which were taken on. First, the field had to prove that it could do something, even something putatively small. This would include checkers, simple chess programs, and the Logic Theorist program, which proved logical theorems. Second, computers at the time had very little space to spare. In the mid-1960s, integrated circuit memory storage was finally implemented and the prospective applications which could be carried out were finally greatly widened. The combination of increased computing capacity, a quorum of participants in the form of both grad students and professors, and a number of increasingly well-articulated research programs meant that there was ‘plenty of meat on the bone’ for research.

The quality of the puzzles and games which were investigated in an effort to embody “intelligence” during this time may seem minute and schematic. How could such problems say anything about particular acts of cogitation apart from tic-tac-toe ? However, suggesting internal strategies in itself was not inconsequential, even if the space upon which such studies were played out seemed

like a small world. The field of cognitive psychology itself, now intensively cultivated, was itself quite novel. The problems were indeed tiny, and did not resemble the larger semantic units which make up the real world. The former minute problems, solved both through computational simulation and through cognitive science experimentation through protocols, could indeed help to show that human beings employed and could even be cognizant of their problem-solving strategies.

The territory to be addressed by AI was generally seen as consisting of the understanding of such small problem spaces. Such small spaces were studied intensively. The analytical surveys of the field in Computers and Thought, the synoptic volume edited by Edward Feigenbaum and Julian Feldman in 1963 (5), devote a great deal of attention to 'machines that play games' and 'machines that prove mathematical theorems'. The games include chess and checkers (6). However, playing such games was not seen as an end in itself but as a fine, limited model in terms of which to begin to map out the way that people went about solving problems.

The term "people" is used deliberately here. A wider view of intelligence encompassing the humble survival skills of insects and mammals would be taken up only decades later. At this time, the human mind is seen as the ultimate measure of all things, including things computer-related. The work in chess and checkers was done prior to 1960- although chess continued and continues to be a subject of AI research. But after 1960, the first applications with more substantive problem spaces appeared, including James Slagle's program for solving calculus problems (7).

In the late 1950s, Newell, Shaw, and Simon ('NSS') had initiated the General Problem Solver program, or series of programs. While they were attempting to find heuristic means to characterize the traversal of a larger state space, GPS did use toy problems. As Minsky characterizes it:

" The most central idea of the pre-1962 period was that of finding heuristic devices to control the breadth of a trial and error search." (8).

However, NSS worked as cognitive psychologists as well:

“ The second important avenue was an attempt to build working models of human behavior incorporating or developing as needed specific psychological theories. In requiring the machine’s behaviors to match that of human subjects, one need be concerned only with overall schematic strategy, for replication of complete individual details would make no more sense here than it would in psych itself.” (9).

Meanwhile, late in the 1950s John McCarthy was beginning to write the LISP language, which would expand AI’s possibilities for computer languages with declarative semantics and hence the ability of the field to express- well, facts about stuff. McCarthy was also developing ideas about how timesharing would work- both at MIT and starting in 1961 at Stanford. Marvin Minsky was continuing to refine and rewrite his overview of the field- which he had actually started following the 1956 Dartmouth Conference. Minsky was working with his first group of graduate students, on the topics of semantic problem solving and natural language issues, input-output improvements, and the earliest mechanical hand work.

In addition to the material impediments of computing in the earliest years of the 1960s, other features of the intellectual topography are worth observing. The inner circle of AI’s founders was characterized by ‘familial’ differences as much as by their fundamental agreement on the possibility of the field itself. These differences, clear even in 1956, were creative and helpful rather than destructive, because of the existence of several different institutions at which the field could progress along different lines. These included, as we will see in some detail later, AI as an artifact demonstrating cognitive psychology’s insights carried out at CIT; AI as engineered artifacts without necessary reference to cogitation, carried out at MIT; and the development of languages, timesharing, and semantics for formal languages carried out at MIT and then at Stanford by John McCarthy and others.

Finally, beyond the purview of AI and its practitioners strictly speaking, other issues challenged the field as it was; these would help to broaden it over the course of the next decade or so. Early

robotics, such as the robotic arm program being carried out at MIT, were within the ambit of the founding of the field. However, this field would instigate the observation that intelligence was multi-faceted rather than being entirely located in IQ-type test problems. This was something that AI's founders had never denied, but it would lead to more explicit claims as to the value of a wider sort of intelligence. A great many engineered artifacts were undertaken even in the earliest days at MIT's lab. This would help to seed the concept of emergent functionality, intelligence without representation, and embodied intelligence, in the 1980s. Oliver Selfridge, an early AI proponent, was developing the Pandemonium program, the first appearance of daemons and agency.

Even as early as 1960, the focus on the engineering of machines which learned from or did the same thing as the physiology of humans led to the invention of the concept of cyborgs, that is a 'cybernetic organism', or a human being who incorporates exogenous parts into its body- engineered parts to give it better vision or locomotion (10). At that time the concept was practically science fiction, but soon enough it would become a practical engineering concept. By proposing an impossible but tangible frontier, the field's proponents and those further afield were preparing to extend that horizon.

The State of the Art in Computing in the Early 1960s

AI's status quo early in the 1960s must be seen in the broader context of computing at this time. This was not 2002 or 2012: computers were rarities, not even depicted often in movies. Applications programs for frankly frivolous pursuits- Angry Birds, mobile restaurant reviews, idle gossip- were unimaginable. The lack of memory available to programs impeded what could be done, even at the Massachusetts Institute of Technology. As Minsky recounted in an interview with computer historian Arthur Norberg:

“ **Minsky:** What was the limitation of AI research? In '58 or '59, the limitation was that the 704, the IBM machine, was more or less fast enough for what we wanted to do, but it didn't have enough memory. By 1962, the programs were starting to occupy two coreloads, which means you would have to run something and then

re-load and run something else and so 32,000 words of memory were beginning to be a pain.

Norberg: How did the new hardware change the nature of the problems that were being worked on?

Minsky: It just meant that you could write bigger programs. I think Joel Moses and Bill Martin started writing MACSYMA in '65 or '66 and those required the biggest machine available because MACSYMA was a huge LISP program that would hardly fit. What we did then was we went and spent another million dollars to buy more memory for the PDP-6." (11).

Explaining these limits in terms of specific components of a computer also indicate both how limited access was for the purposes of AI research, as well as how prescient AI itself had been in its foundation. The basic components required for a digital computing machine, electronic or otherwise, from the abstract designs of Charles Babbage through the relay machines and up to the era of digital electronic computers, are logic, memory (the store in Babbage's terms, or registers in earlier machines), storage or external memory, software, and forms of input and output (I/O). Any improvements in the effectiveness of computing overall would come about from improvements along several different frontiers. Perhaps more than had been initially understood from the promise of digital computing, this proved to be a great deal of work. These different components improved unevenly, but in some cases dramatically, from the end of the 1950s throughout the 1960s. We will briefly but specifically indicate how in this segment then in the next segment, indicate the deliberate means by which ARPA encouraged advances in computing.

The sophistication of the earliest AI stands out in contrast to the cumbersome nature of the facilities in which it took place. Like a neonate born before its forty gestational weeks have passed, AI as a concept was present in the world before the massive computing infrastructure necessary to easily [sic] carry it out had become available. The bones had not yet calcified into hardware, so to speak, which used reliable and rapid components; there was almost no software to facilitate respiration.

The Logic Theorist, we may recall, was actually written in 1955 and 1956 in IPL, the computer language which had been written by Newell and Shaw, was almost directly written in the machine language of binary signals. Late in the 1950s, and several years later with the appearance of the IPTO, there was concerted demand for more rapid and immediately accessible computing power. But generally a preponderance of those who knew computers did not advocate user-friendliness but were instead concerned with reliable components. Today the idea of a computing machine as big as a school bus, protected by air conditioners guarded by operators who ran the batch programs and kept the curious away, seems odd and comical. Yet it was a great improvement over the past fifteen years of computing found only in military and a very few corporate locations. Since the world population of computers did not exceed six thousand or so (12), usability or the opportunity for individuals to present their programming directly to computers may have seemed like an extraordinarily audacious demand.

Input and output were a ponderous and protracted affair, involving the preparation of punch cards or tape and a long delay, sometimes several days, while these were put into a batch with a number of other programs and put through the machine. Because machines could read the tape or cards faster than humans could prepare them, it was generally thought that this was the permanent status quo. Punch cards remained predominant, if not optimal: as the '50s ended, two-thirds of IBM's income came from leasing punch-card machines (13). The dynamic memory (registers and accumulators) was small, and storage itself was often considered to be as well embodied in punch cards as in the magnetic media. Some of these problems were solved by improvements such as timesharing and the design of buffer zones to hold input and output for timesharing and for better access to peripherals. But well into the 1960s neither of these obstacles was thought of as a problem. Many discretionary user-friendly improvements, such as the introduction of timesharing, will be examined in the next segment. Instead, in the 1950s these improvements were almost invariably to the power and reliability of the arithmetic and logic units.

The digital computer's original vacuum tubes were themselves almost immediately superseded by the invention of the transistor, a

small and intricate binary state circuit formed of [small] blocks of various elements. (Electronic technology using semiconductor materials in block form is called solid state for this reason). While the transistor was invented shortly after the end of WWII, it was not ready for commercial implementation in computers or other devices until Bell Labs patented it and then began issuing production licenses in 1952 (14). During this time, vacuum tubes were used for computer design while the transistor underwent R&D by Bell Labs and the Army Signal Corps (15).

Solid state electronics using the transistor appeared in the portable radio in 1954, resulting in a tremendous commercial success (16). However, it was only late in the 1950s that transistors were first put into two military-market products, the latest version of the UNIVAC, and a computer manufactured by Philco and called the Transac S-2000 (17). This was followed by its implementation in many other computers, most notably the IBM 7090, which was the one most often leased to universities. The introduction of transistorized logic resulted in a computer which operated five times as fast as the IBM 709, and did not need special air conditioning or allowances for immense usage of power (18).

At the time that our story starts, all computers were mainframes and were still powerful in direct rather than inverse proportion to their size. These facts still hampered all Computer Science research by their adherence to Grosch's law of increasing returns to scale rather than Moore's law of increasing returns to smaller processing unit size. As we will see in the next chapter, the commercialization and intense military need for the Integrated circuit were about to change this, but the fact remained.

3. Objections to AI at the Start of the 1960s

If one essential challenge to the field of AI as of our time period was technical in the strictest sense, the other was cultural. Anti-computing arguments which had been faced by Alan Turing early in the 1950s, and those faced by Paul Armer, Director of Computer Sciences at the RAND Corporation, a decade later, do not differ greatly. This is not surprising because of situation with computers relative to the general public and the university and cultural

audience in 1950 and in 1963 likewise did not differ so greatly. In 1950, only a few digital computers were to be found in the whole world. The Eniac, the first working digital computer, had indeed been unveiled to the public, several years after its birth. But thinking machines were only beginning to be seriously contemplated. When Armer wrote in 1963, the IBM 360 project, which would greatly increase the public visibility of computers, was underway and timesharing, teletype-and-monitor I/O, were still confined to a very few military and research facilities. Computers were still confined to scientific and research and closely controlled corporate facilities, and apparently had not developed a much higher public profile. The avalanche of appearance of these machines was definitely on the way, but not widely visible (19).

Repeated jabs against 'thinking machines' speaks to distrust of their possibilities. The 1963 compendium Computers and Thought includes Armer's literature review, "Attitudes toward intelligent machines",. Armer fends off a flurry of spurious jabs, many of which originally appeared in Turing's "Computing Machinery and Intelligence" a dozen years before (20). Both Armer and Turing point out arguments by stipulation, that is ones which simply assert, "Let's settle this once and for all, machines cannot think!" or "A computer is not a giant brain... It is a remarkably fast and phenomenally accurate moron" (21).

An extreme example of such stipulations is the 1961 work of Mortimer Taube, which lumped AI together with some of history's greatest scientific impossibilities:

"1) is it possible to translate by machine from one language to another?...5) - is it possible to have extrasensory perception?... 10 - is it possible for a machine to think?" (22)

Turing and Armer both also observe extremely weak arguments such as the theological one- "Thinking is a function of man's immortal soul" (23), and the 'heads in the sand argument'- 'The consequences of machines thinking would be too dreadful. Let us hope and believe that they cannot do so.' Armer also cites "the argument of superexcellence", which includes statements that computers surely can never compose music as well as such singular

human talents as Mozart and Chopin (24). Armer also reiterates Turing's observation of an attempt to simply define AI out of existence by asserting that intelligence per se requires biological life as humans know it (25).

Another source of attacks against computers more generally- and against AI only by derivation- came from prominent engineer and Cybernetics founder Norbert Wiener. He asserted the moral weakness of building computers to be slaves (26). This is surprising, given that most of the arguments against computers and computing per se, and are evidence of the general level of ignorance found even in the educated publications of the day. Wiener could hardly have been accused of such ignorance. Usually the arguments assert that computers cannot remember enough material to be useful (27).

In a 1963 sociological study, the two most commonly noted images of "electronic thinking machines", that is computers, were of "beneficial tools of man", and "awesome thinking machines". Some of the corollaries of the latter image indicate magical thinking and attributions of psychic powers, rather than mere data-crunching, amongst such machines. The statements illustrating this perspective include "someday in the future, these machines may be running our lives for us", "There is no limit to what these machines can do", and "They can think like a human being thinks". The facts of the computing scene relative to the general public had indeed changed during the 1970s. But even twenty years later, some of these beliefs appeared in the survey results, although they were held by many fewer people (28).

As late as 1962 and 1963, Newsweek magazine referred to computers as "mechanical brains". Perhaps such terms were the intermediate steps in a long process of understanding of digital computers in their own terms, finally independent of Cybernetic images of the human mind. Both because of ignorance and because of misplaced offense at the endeavor of AI itself, the essay was bound for criticism.

As it would do later, AI acted as a lint brush for the ambient anxiety of the age. The general anxiety was certainly justified, even if the

target was the wrong one. As the Cold War's outlines sharpened, it became clear that no combatant- not the USA, not the USSR- intended to lay down their arms and beat their swords into ploughshares (29). For anyone already anxious about the breakneck development of nuclear weaponry, the idea of thinking machines, prior to the term "AI", must have seemed a likewise appropriate target. The transmogrification of existing nuclear weapons and computers into a terrible beast that embodied both must have seemed possible.

The legitimacy of the fear does not mean that AI was indeed the lackey of nuclear warriors. It was instead one of the earliest pacifistic and scientific purposes to which computers were turned. As discussed in the book preceding this one, Newell, Shaw, and Simon began working together as a result of the SAGE anti-missile defense project.

By its nature, this text will perhaps emphasize the support that the fledgling AI community received, but this was not its sole reception even within the scientific community. The advent of ARPA funding for what were then exotic sciences provided a measure of protection for AI practitioners in terms of freedom to continue their own work. However, through much of the 1960s, AI suffered from a poor reputation among scientists and particularly among computer scientists (30).

Conclusion

Thus, this book commences with a cusp. 1961, the year in which this book begins, may be the very last year in which AI was a fledgling academic field, made up of a dozen or so people. Even in this year it was not marginal, but was carried out by tenured professors in major institutions. With this tiny inner circle and a progeny of another two dozen students at best, AI proceeded at a pace that was not surprising given the scale. Computers, in themselves, were well-funded because of their close connection to weapons information systems. The general trend of computers as such was toward a solution to the storage and heating problem of enormous components and room-sized mainframes. Necessity may indeed be the mother of invention, but in the early 1960s there was

little conception even in research circles of a pressing need for mass production of smaller computers. This was in itself impossible at the time given the propensity to larger rather than small computers.

The integrated circuit had been invented but was not being developed in meaningful quantities. There was inadequate material need- effective demand, as the economist would call it- for smaller processing units to be developed in huge quantities and far more cheaply.

In the next chapter, we will see that very soon, world events would create a demand for every imaginable sort of computing, and this would allow the field to add students and programs and courses and buildings and timeshared systems. The changes that would take place later in 1961 would have a tremendous impact on its funding. In turn, the repercussions of this event would greatly alter the need for AI and every other field of study related to computing.

Part II. Twin Bolts of Lightning Alter the Landscape

Lightning is said to not strike twice in the same place. For the few so afflicted, this is unfortunate. In the instance we are discussing, it was nothing short of a miracle. Some miracles come from Heaven, others from Washington, D.C. AI was one technology blessed by such exogenous (stochastic ?) lightning strikes during the early 1960s. The integrated circuit [IC], invented during the end of the 1950s but not really subject to the demand that would allow it to grow until 1961, is another such miracle that made computing faster, cheaper, and eventually far more ubiquitous.

As we will see in this segment, both hardware advances and institutional support for AI and other esoteric computing applications were necessary. We will address these intertwined developments in two successive chapters. The appearance of ARPA, first as ersatz missile agency and then as computing's champion, was historically fortunate for AI once ARPA's energy was turned toward computing. But ARPA's enthusiastic program directors in themselves did not unleash the flood of computing power that

appeared shortly after IPTO was founded. That took three people and a great deal of history to set in motion: Jack Kilby, Robert Noyce, and President Kennedy. The integrated circuit was invented independently by Jack Kilby at Texas Instruments in 1958 and Robert Noyce at Fairchild Semiconductor early in 1959 (31).

Chapter 2. The Revolutionary Impact of the Integrated Circuit

Prior to the appearance of the integrated circuit, circuit creation was an impossibly complex matter of assembling far too many transistorized components into one unit. Welding them together was so complex that they were almost impossible to build. The 'Monolithic Idea', as Kilby called it:

"...resolved the tyranny of numbers by reducing the numbers to one: a complete circuit would consist of one part- a single (monolithic) block of semiconductor material containing all of the components and all the interconnections of the most complex circuit designs... the tangible product of that idea [was] the semiconductor chip" (32).

The technology is universally referred to as the 'IC' at present, but was also called 'monolithic blocks', or when used for storage during the late 1960s, thin film storage (33). The fundamental substrate of integrated circuits is ceramic semiconductor material. In this way, it differs from ferrite (or iron-based) materials more commonly used for magnetic core memory and storage during this time.

We may be blasé about it now, but the IC and its surrounding technology are marvels of ingenuity:

" The integrated circuit, as we conceived and developed it at Fairchild Semiconductor in 1959, accomplishes the separation and interconnection of transistors and other circuit elements electrically rather than physically. The separation is accomplished by introducing pn diodes, or rectifiers, which allow current to flow in only one direction. The technique was patented by Kurt Lehovec at the Sprague Electric Company. The circuit elements are interconnected by a conducting film of evaporated metal that is photoengraved to leave the appropriate pattern of connections. An insulating layer is required to separate the underlying semiconductor from the metal film except where contact is desired. The process that accomplishes this insulation had been developed by Jean Hoerni at Fairchild in 1958, when he invented the planar transistor: a thin layer of silicon dioxide, one of the best insulators known, is

formed on the surface of the wafer after the wafer has been processed and before. The conducting metal is evaporated onto it.” (34).

The first commercial IC, produced by Fairchild, was greeted with trepidation at its first industrial exposition, at the IRE convention in 1961 (35). Yet demand soon appeared in the form of politics. The same year, President Kennedy announced Project Apollo, which would later put a man on the Moon. The announcement of the NASA mission precipitated immense need for smaller computing components, and indifference was replaced by rush orders. The U.S. government provided one hundred percent of the market for integrated circuits until 1964, and much of the market for several years after that (36). The integrated logic units were used for the Apollo program, but not only for that. Early ICs were put into various uses in Navy programs and in NASA satellites (37). Volumes of IC production and the individual price per unit skyrocketed and dropped drastically, respectively. (Politics does indeed make strange markets, in addition to strange bedfellows. Only the former and not the latter appear to have applied here). The development thus subsidized by military programs was critical to the prices of the chips dropping enough to entertain a wider commercial market (38). These markets were considerable indeed: over a decade IC sales rose from almost nothing to \$130 million (39). Demand, combined with certain sales, meant the rapid development and improvement of the technology over a period of a dozen years. The torrential flood of integrated circuits being produced and let loose on the world in more and more computers produced positive feedback in the sheer juggernaut of faster computing. According to Robert Noyce:

“ After the introduction of the integrated circuit in the early 1960's the total world consumption of integrated circuits rose rapidly, reaching a value of nearly \$1 billion in 1970.” (40).

The Appearance of Moore's Law

Nor was there an end in sight. In 1964 Gordon Moore, then director of research at Fairchild Semiconductor, saw that the number of elements (roughly speaking, the complexity) in integrated circuits had been doubling every year since their invention in 1959. Moore's

Law, the dictum that “the number of components per circuit doubles every year’ (41), was born. According to Moore’s Law, this trend will go on; its limits certainly were not reached during the 20th century.

The progress in IC continued with a widening of their applications even during the 1960s. Increasingly successful throughout the 1960s, integrated circuits used for logic circuitry became one of the characteristic regional specialties of the Silicon Valley. In contrast to the giant think tanks attached to universities and corporations on the East Coast, the firms in the Silicon Valley relentlessly engaged in fission and splintering even as they grew, often spinning out a new line of computer technology with each new firm. At this early stage, production itself as well as research and development took place in the Valley (42). Shockley Semiconductor Laboratory, founded by the Nobel-winning scientist (and avowed racist) himself, quickly begot Fairchild Semiconductor, at which the IC was invented (or more precisely, co-invented). In 1969, Noyce and Moore left Fairchild Semiconductor and started their own company, Intel, in nearby Santa Clara. Their particular angle on ICs was the introduction of miniaturization to data storage. As we saw earlier, nature abhors a vacuum, or in this case an old technology. Internal memory was still implemented in the form of magnetic cores. The new chips which Intel introduced in its first several years were called memory chips. This product was immediately in fierce demand, and the unit price of computer memory (for the 1103 memory chip) dropped as production skyrocketed (43).

Several dozen competing firms, all initially composed of defectors from Fairchild Semiconductor, were founded in addition to Intel. The fantastic flood of hardware assured computer users of no end of opportunity. As we will see when we examine hardware developments in the 1970s in Chapter 12, the microprocessor would change computing at least as dramatically as the integrated circuit had.

Chapter 3. The Revitalization of the Advanced Research Projects Agency and the Foundation of the IPTO

The birth of Classical AI was, as we have seen in the last two chapters, the product of some intensely nurtured and echt-Classical strands of Western civilization. However, this grand structure was indebted to massive and harsh forces, such as governments and Departments of War, and university laboratories of basic science. To understand why AI was able to live long and prosper, rather than simply subsisting as a highly academic field of applied cognitive psychology carried out by Newell and Simon and perhaps two dozen others, we have to walk down the dark alley that leads from the end of the Second World War to the grim illuminated stage set of the Cold War. The Cold War is brightly lit, but strewn with human tragedies. This is not true, however, of the technological marvels which were called into being by the Furies of the Cold War. AI was one such technology, although it was not perceived as one which was very important, either when it started or for a half-dozen years afterwards. The SAGE Project was the one which specifically inspired AI, but this was only one of at least a dozen. Eventually, the Cold War generated a technically imperative project which allowed, and energetically encouraged, AI to flourish.

In the previous chapter, we had to backtrack slightly in history to explain the significance of the IC to AI and to computing more generally. We will likewise have to backtrack to see how Sputnik and the Apollo Project, and the subsequent appearance of ARPA's IPTO were the alternate keys to the new reality of computing in the 1960s.

The Spark of Sputnik

"Space, the final frontier". Gene Roddenberry.

On October 4, 1957, the Soviet Union sent SPUTNIK I, the first man-made earth satellite, into orbit around the Earth. It remained in orbit until January 1958. The Soviets followed this foray with Sputnik II later in the month. The alarm with which Sputnik was received by the United States suggests that the satellite was launched into space out of the blue, on the sudden whim of the Soviet leaders. This is not at all the case. The respective high-ranking name-callers of the two superpowers' governments (Soviet Premier Nikita Khrushchev and U.S. Secretary of State John Foster Dulles), had

been blustering at each other for much of the decade, while developing intercontinental and space missiles which threatened to make existing distant early-warning projects (such as SAGE) obsolete.

It is perhaps only coincidental that the Soviet Union narrowly beat the USA in launching a missile in space. Nominally, the pursuit of military aims in space, originated in a corruption of the International Geophysical Year, a scientific surveying project. Surely the original idea of the IGY had not been as pure as the driven snow, politically speaking, to begin with. But much of its scientific concerns were hijacked by the international arms race.

The United States went ballistic, so to speak, and responded with accelerated development of its own projects, the Thor and Jupiter missiles. There were eight more attempted satellite liftoffs in 1958-1959 alone by the both parties. Whatever the larger context of the ongoing arms race, the Sputnik missile launch permanently turned up the volume (44). But the U.S. response was broader than simply stepping harder on the toes of the engineers at Cape Canaveral, Florida. By mid-January of 1958, President, and former General, Dwight Eisenhower, had established a bureaucratic separation between the missile projects and other divisions within the Defense department. This could be construed to have been done so that internal conflicts of interest within the Army and Air Force- both of which had missile projects- would not impede more dramatic progress on these missiles. To this end he established the Advanced Research Projects Agency, to answer directly to the Secretary of Defense, "for the unified direction and management of the antimissile missile program and for outer space projects." (45)

Over the next calendar year, President Eisenhower took other actions which strengthened the scientific cohort within the military. He appointed James Killian, who had been the President of MIT since 1948, as a presidential assistant for science. Killian in turn immediately established the President's Science Advisory Committee (46). Eisenhower also set in motion the creation of the position of Director of Defense Research and Engineering. This position superseded the existing one of Assistant Secretary of Defense for Research and Engineering, and provided further jobs

within the military-intelligence-computing establishment. Later ARPA executives, for instance George Heilmeyer, sometimes started in DDR&E. Congress also began paying more attention to space sciences; the Senate established a Standing Committee on Aeronautical and Space Sciences. ARPA had been placed in charge of missiles in specific and upper atmospheric exploration and outer space vehicles in general, but this lasted less than one year. Its specific administration of missile projects was complemented and broadened by the foundation of NASA. A government body which carried out NASA's functions already had existed since 1916. The National Advisory Committee for Aeronautics, or NACA, was the proto-NASA, just as the ONR was the proto-ARPA. NACA was renamed and given \$125 million and a far more prominent position in 1958.

Sputnik was manna from heaven to the United States' scientific community in general. The response was vastly increased spending on military research and development, especially where scientific education and experimentation were concerned. The United States was jolted into a frenzy of military and scientific spending:

“ The nation responded by authorizing billions more in R&D- specifically, expenditures went from \$3 billion in 1957, the year of the Sputniks, to \$15 billion in 1964. Research aid to universities spurted upward during this period, as did budgets for research into all forms of education.” (47)

We must backtrack chronologically in order to see the significance of the origins of ARPA. ARPA and NASA were not immediately consequential for AI in its very earliest years, because the technical needs of aeronautics did not, at first, seem to include intelligent control. It was several years before a broader defense agenda- possibly the broadest scientific agenda ever implemented by any government- saw a need to foster 'exotic' computer research. Between 1958 and 1962, the Agency functioned as an "interim space agency", settling by the early 1960s on the goal of developing re-entry physics for missiles (48). Jack Ruina, ARPA Director from 1961 to 1963, stated that “The major programs we had were ballistic missile defense [and] nuclear test detection”. Other things were on the margin of the larger defense picture (49).

The turn toward governmental support for AI, and a number of other sensory emulation technologies, was the result of a sequence of events, most of them over the course of 1961. As we saw earlier, integrated circuits had found no buyers when Fairchild Semiconductor first tried to sell them in 1961 (50).

Here the story of the arms race rejoins the story of the IC. But history intervened in a way that would jolt every aspect of the commercial and federal systems markets for computing. We have the Soviets and the arms race to thank for this. As the USSR sent astronaut Yuri Gagarin into space, the United States responded by stepping up its existing discussions of human space flight. U.S. President J.F. Kennedy, never one to back down in a contest of machismo, responded by launching a person into space the next month, and raised the stakes by proposing the human space flight to the moon (51). Instantly, NASA was even more exciting than it had been, and there was, moreover, an instant and insatiable market for integrated circuits. The Apollo program as the Nasa moon shot was called, took eight years and uncounted millions to achieve, and provided a considerable market for IC's. Precisely, it made up 100% of the market for integrated circuits until 1964. For most of the rest of the 1960s, the federal government remained the biggest buyer of IC's (52). Because of military demand for IC's for the Apollo program (the NASA Moon shot), for Navy programs, and for NASA satellites, production of the chips surged and prices dropped (53).

The invention of integrated circuits, and their commercial improvement by the United States government, made all forms of computer applications cheaper, and therefore easier to endorse. This supply-side, or technology-push, invention was complemented by another major development on the demand-pull side, again independent of AI. It should come as no surprise that the capacity of the DoD to develop space-exploration projects would eventually be impeded by the limits of its own software and programming capacities. Norberg and O'Neill indicate that there was:

"...a recognition inside the DOD of shortcomings in command and control systems. By 1960 the DOD recognized that it had a problem with respect to large amounts of information requiring timely analysis, and DARPA received the

task of examining how to meet this need. Command and control denotes a set of activities associated, in military contexts, with rapidly changing environments. These activities include the collection of data about the environment, planning for options decision making and the dissemination of the decisions. Increasing the amount of strategic information that could be controlled would improve the command decisions needed in a rapidly changing environment; and computing technology had the potential for controlling greater amounts of information and presenting it in effective ways to aid decision making.'

...Earlier military concern with the processing of information and the use of the results in command decisions had focused on the problems of human relations. In the early 1960s, however, the focus shifted to the informational aspects, and computer use in military systems expanded..." (54).

The Computer's Use for the Space Mission Begins

In 1960, ARPA requested that the Institute for Defense Analyses analyze the prospective contributions of information sciences, broadly understood, to defense. IDA suggested that information science could contribute to defense in the areas of 'pattern recognition, decision making, communications, control, and information storage and retrieval, data handling and data processing'. This list could include a sufficiently wide swathe of all exotic computing- Cybernetic, AI, and otherwise- then being done that the funding of 'new' research by ARPA would allow many marginally relevant technologies under the military umbrella later. Decision-making, which Newell and Simon were already pursuing as psychologists, was perhaps the most closely proximate to command and control. Early in 1961, President Kennedy complemented the metal-and-plastic side of the Apollo mission with an equally necessary initiative in command and control. The IDA report the year before had not been implemented, but it was taken up now. ARPA gave the Systems Development Corporation, the Rand spinoff at which NSS had first envisioned Complex Information Processing, millions of dollars to "pursue research on the conceptual aspects of command and control systems" (55).

In June, 1961, DoD Secretary Robert McNamara issued a more specific directive, mandating research in Command and Control, or "C2". This term usually refers to logistical control exerted over weaponry deployment. But in this instance, C2 devolved to research that had no immediate military objectives, or even stated

prospective technological outcomes. ARPA established the Command and Control Program in 1962, and changed its name to the Information Processing Techniques Office soon afterwards (56). As IPTO's first director, Ruina appointed MIT acoustics engineer and psychologist J.C.R. Licklider.

The IPTO mission, and the larger ARPA mandate, was purposely open-ended and broad, allowing scientific minds to wander fruitfully on ARPA's tab during this period. Not everyone accepted this: even by the early 1960s, basic science faced the habitual distrust from the armed services and 'middle American' society- and the positive loathing of the intellectual avant-garde. Fortunately in this specific instance, the governmental agencies were relatively shielded from public opinion.

The IPTO and ARPA initiatives established what Uncle Sam wanted from AI. For at least a decade, this was blue-sky research which would strongly aid the Space program, and which would also clearly enhance military intelligence objectives (57). During the decade and a half between the foundation of the Advanced Research Projects Agency (1958) and the traumatic conclusion of the Vietnam War (1974), AI's aspirations and those of ARPA closely coincided. Like all good things, this one came to an end, but not for more than a decade, and not before substantial work had been accomplished.

ARPA and the Early Primacy of the Major AI Centers

"All roads lead to Rome."

Artificial Intelligence is no more an orphan institutionally than it is intellectually. It has historically been dependent on the support of ARPA in specific, and the Department of Defense more generally. For decades, ARPA was quite the nurturing parent. Many roads in AI lead back to ARPA, a branch of the Department of Defense and a silver lining in the post-war military cloud. The IPTO has underwritten much of the research discussed in this book (58). Even before ARPA and IPTO, the people who would later become AI's founders, and their first students, were supported by grants and contracts and jobs at a dozen or so top Cold War institutions- the Office of Naval Research, BBN, Lincoln Laboratories, the RLE, Rand,

the Air Force Office of Scientific Research, SRI, and the major universities themselves. But ARPA provided an ideal basis for support for this and other fields of research.

The IPTO was AI's funding agency for three decades, and was responsible for practically all AI monies during this period (59). During the 1950s, ONR monies to Newell and Simon and for various meetings had been consistent but small-scale. In contrast, IPTO monies were consistent and considerable. Beginning in 1962, IPTO gave its imprimatur and funds to a number of institutions. The major centers for AI included MIT, Stanford, and the Carnegie Institute. Yet this is not the entire list. The larger list included MIT, Stanford, SRI, the Systems Development Corporation (in San Diego), the University of California at Berkeley, U.C. at Los Angeles, and U.C. at Santa Barbara, the University of Southern California, the Rand Corporation, Carnegie-Mellon, and the University of Utah. A list of grants given to individual research projects would be more extensive, as would a list of institutions which participated in the early ARPANet email system.

Increasingly, ARPA enforced the bias of scientific funding toward California. Secular trends during the postwar decades through 1990 included a steady increase in funding, the addition of at least a dozen research centers primarily dependent on IPTO money (during the 1960s and early 1970s), and an increased skewing of funds to universities (in the 1970s). As we will see in Chapter Eight, the mid-1970s marked the nadir of PostWar military spending. However, IPTO subsidies contradicted this tendency, as even more AI programs were initiated during the late 1970s and the early 1980s. These included the California Institute of Technology, Rutgers, and the Universities of Rochester, Maryland, Massachusetts, and South Carolina (60). Such funding was unwavering and would even increase drastically in real terms during the stepped-up military spending of the 1980s (61). As we will see in Chapter Eight, military grants to AI were indeed threatened during the contraction of the military-industrial complex in the early 1970s, and the habitual beneficiaries complained bitterly when the terms under which they were given grants became more stringent. Yet for most research purposes, AI ultimately got its cash. A dramatic drop in funds to IPTO (ISTO, SISTO) under its new auspices as NIST in the 1990s

simply illustrates the direct relationship between military funding and monies for advanced computing research (as opposed to commercialization) which has prevailed since World War Two.

The IPTO and ARPA Managerial Style

Cost-plus contracting, as the distinctively expensive administrative model of the military-industrial complex was known, was far from the most economical way to conduct the Cold War in the United States (62). Almost all of the actual research and development work was parceled out to major weapons manufacturers. Cost overruns were routine and often exorbitant. It is not surprising that it has been widely criticized.

In contrast, IPTO's managerial style differed greatly, and the distinction proved to be fortuitous. Indeed, the agency was distinctive in that all research was contracted out to universities and the research divisions of major corporations. Unlike the Armed Forces per se, the Agency's own staff was skeletal and its costs modest. Moreover, the university contracts typically ran far more modestly than military contractors, and never incurred the chagrin that the latter faced when public sentiment toward war in general turned sour.

IPTO directors, almost all research scientists recruited from universities, were given free rein in determining which projects to accept. Beyond this, decision-making regarding fund allocation was often turned over to the participating scientists themselves. J.C.R. Licklider stated that the military overseers above him in the DoD were largely ignorant of the exotic work funded by ARPA:

" At that time the high-level administrators ...did not have even the vaguest knowledge of AI and so my charter was essentially 'Computers for Command and Control'...it was my position- and one I was able to defend- that if anybody needed artificial intelligence, it was the Department of Defense... Fairly early, ARPA got into the idea of asking participants members of the research community to try to design new projects. This led to a series of annual PI (principal investigator) meetings... I would guess that the ARPA community is responsible for originating at least half of the ARPA programs. Probably what the Pentagon thinks the DoD needed is responsible for about 30 or 40% of the programs" (63).

Robert Bartee of the Office of Naval Research, a prominent administrator who had helped to support the Dartmouth Conference, believed that the main role of the grants administrator was to find smart people and give them the resources required to do their work (64). The IPTO took the same philosophy, acting as an open-ended subsidy to individuals as much as to their projects. This meant that the individual project managers played an active role in finding people. Since they were recruited from the community itself, and knew where to look, this was not too difficult. J.C.R. Licklider in particular is widely credited for this. His academic life had trained him perfectly for making the choices his job required him to make. He crisscrossed the United States pursuing promising computer scientists. He sought out Newell and Simon and their students as well as people he knew from MIT- Newell, Feigenbaum, Minsky, McCarthy- and asked them to lead projects, and solicited computing projects before the researchers asked him. IPTO's initial projects consisted of timesharing at MIT (Project MAC); computing infrastructure at the Systems Development Corporation and at the University of California at Berkeley; improvements of computer displays at the Stanford Research Institute; John McCarthy's research at Stanford; and the AI work by Newell and Perlis at the Carnegie Institute of Technology (65).

ARPA's project managers knew where to look, because they were part of the community, and because, at first, that community was concentrated in a few pinpoints on the map of the United States. ARPA and IPTO program managers appear to have been drawn from MIT in particular. Many of the early ones, including Larry Roberts, Ivan Sutherland, J.C.R. Licklider, and Robert Kahn, were MIT graduate students or professors, or both (66). Typically in academia and science, peer review is held out as the best means by which to assure objectivity. But blind peer review was not a possibility for AI, since the field was simply too small. With barely two dozen 'peers', information was inevitably transmitted by personal contact.

The administrative model which ARPA and the IPTO followed during its first decade seems to have been light-handed in that it did not simply commission products but encouraged the formation of new science. But this modus operandi itself was still more active than

one would think. It substantively influenced AI by what it did and did not require of its beneficiaries. The form in which IPTO administered grants left much decision-making up to the individual professors and lab managers. The grants compelled people to finish their projects, in order to have a working computer program to present. But ARPA grants did not require that work be tested in an industrial setting, or that it be useful outside of 'science'. The university laboratory, and carefully-staged presentations for grant providers, sufficed to keep funds and academic prestige flowing. Given both the nature of the market (insofar as ARPA was the sole consumer) and the early state of the science, it is not surprising that there were no commercial applications during the 1960s. This state of affairs did not last indefinitely, and indeed ended during the waning days of the Vietnam War.

The circumstances were nearly unique to the time, and helped to create an intellectual environment of unparalleled energy and vitality. Science does not live by ideas alone. There are numerous encumbrances: grants applications, conferences to be attended, results to be presented and presented again, organizational meetings, grants-presenting bodies to be placated, departmental infighting, mucilage and paper and documents and documentation. But much of the hurly-burly of scientific and academic politics was suspended during this time. The ease afforded the AI scientists a unique concentration on research. Today's research scientist, however talented, must excel in grantsmanship [sic] in order to survive. The professor at the height of the Cold War did not need to bother with such prosaic things as meticulous, lengthy research proposals, or one-year waiting times to hear about the reaction to those proposals. The formal requirements for procuring research money were often as informal as picking up the phone and calling an old colleague down in Washington. Today's scientist, whatever the scientific field, is almost always casting his or her eye toward possible support from industry. But none of the early AI people had to be oriented toward industry. They worked at universities, in the research laboratories of major corporations, or for the government, during their entire professional lives. This, too, began to change during the 1970s; but perhaps that period in itself defines the generational shift (67).

The Other Benefactors

ARPA was not the sole source of support for AI. Who else wrote checks ? Newell and Simon successfully tapped a number of faucets over the years, including Rand, the Carnegie Corporation, and during the late 1960s the National Institute of Mental Health (68). Nor was ARPA the very earliest source of support for AI, as AI. The ONR, the Rockefeller Foundation (for the Dartmouth Conference), the AFOSR (Air Force Office of Scientific Research), various IRE, ACM and Joint Computing conferences, Rand, various block grants from the DoD (e.g., JSEP grants), and the Carnegie Foundation (at CIT), contributed to AI during the 1950s (69). The Rand Corporation paid in Newell's salary for years as he worked remotely, in Pittsburgh—doubtless one of the first people to have done remote research computing through dedicated telephone lines. Rand also sponsored summer conferences for cognitive psychologists, organized by Newell and Simon, in 1958 and 1962.

Interdisciplinary projects were funded from various pockets. The National Science Foundation was apparently not the salient source, simply because its grants were too small (70). For the early Stanford research, some of which was conducted in conjunction with the Medical School, the National Institutes of Health was crucial in providing for expensive computers and computing time (e.g., for the Dendral project, as covered in Chapter Six). Late in the 1960s, the computer network funded with NIH was made more widely available by hooking it to the ARPANET, with J.C.R. Licklider's permission. The medical computing facility subsequently became Stanford University Medical Experimental Facility for AI in Medicine, or SUMEX-AIM, a prominent software repository. NIH continued to fund AI research, in medical and scientific production systems and other topics, on the SUMEX-AIM facility through the early 1990s (71).

3. ARPA's Contributions during the 1960s

The role of ARPA, and that of the IPTO, was largely one of encouragement of existing agendas. It encouraged the existing projects of the first generation of AI people by granting the individual researchers the grants for their direction of projects, and

by bringing them together at yearly PI conferences. Later, students had their own conferences. ARPA provided money for student budgets, money for more and better computers and for timesharing and later networking in particular. Hence more students entered the existing programs, and the first generation of graduate students entered new programs at Stanford, Berkeley, and MIT. Those students could pursue their projects, rather than being discouraged and finally driven away by not having access to the machines they needed. ARPA grants were generally loosely matched to projects many years in duration, although in the case of CMU the funds went to the center as a 'Center of Excellence' rather than for a specific project.

With the exception of time-sharing and networking, the ARPA-IPTO administrators did not push ideas on people. Had they been interested in directing people, this was not the right group to try to direct. As Stanford/ SRI roboticist Nils Nilsson said later of the period in the 1970s when ARPA did try to push, "I don't take redirection too well" (72). One might as well try to herd cats. The IPTO seems to have mirrored some of its leaders' personalities, fortuitously so for the encouragement of ideas. Licklider was reportedly so mild-mannered that he did not become angry when others claimed credit for ideas he had just expressed in meetings (73).

Like good stereo equipment, ARPA amplified the existing sounds. But the agency did more than turn the volume from low to high. The enormous influx of monies, combined with the great price drop in hardware and the great increase of availability of timesharing throughout the early 1960s, dramatically changed the scene. Larry Roberts, later Director of ARPA-IPTO, remembered computing time as being available only to a few people at MIT:

" Around 1960, [computer science] virtually did not exist as a subject, or as an activity. They did not have computers. The 704 was the Computation Center's only utility. If you were one of the circle of people that worked on WHIRLWIND, fine, you worked on WHIRLWIND. If you used the 704 you were probably in a white jacket and worked for IBM. Otherwise you submitted programs from somewhere and where no students had access to them." (74).

This scenario changed when MIT proper brought in a TX-0 computer, made it accessible to undergraduates, allowed teletype-

and display screen access, and then, with three million dollars granted by ARPA to Project MAC, implemented successive waves of timesharing systems.

While we take it for granted today, given that it is implicit in all networked computing programs, this is not an invention one should approach as an ingrate. In this programming artifact, two or several or, well down the road, hundreds of users could program at the same time, each apparently given individual access to the central processing unit. It was developed through many successive and sometimes concurrent projects. Project Mac was one of the most aggressive in targeting timesharing, and the last of its MIT campus timesharing systems allowed several hundred users access to the computer at a time. An adequate quantitative change will lead to a qualitative change as well, and countless software programs emanated from such grants.

The tremendous quantitative changes, in the form of the influx of the green stuff, helped. But of course the ARPA IPTO leadership did more than write checks- however much discrimination it exerted when it wrote checks. The IPTO leadership, Licklider in particular, was enamored of the idea of computational infrastructure, in the form of 'man-computer symbiosis', the computer utility', and timesharing. Within several years of the foundation of IPTO, and largely due to endless iterations of frustration at the technical impediments to sharing programs and data, the Arpanet was well underway.

Thus ARPA was closely involved with the entire oeuvre of AI development at CMU, Stanford and the SRI, and MIT, which we shall consider in the next three chapters. However, as Licklider made clear in his writings, his vision of the usage of computers was not mainly an AI vision. This does not mean that AI did not benefit, as the volume of work and number of people involved increased from several dozen working with NSS at CIT to much more in many locations.

Licklider and the Quest for Man-Computer Symbiosis

J.C.R. Licklider's central quest was for "Man-computer symbiosis", a concept which he referred to well before he arrived at IPTO. "Man-

computer symbiosis”, or “integration”, is also referred to as “procognitive systems”, and even the closely associated term, “the Intergalactic Computer Network” (75). “Man-computer symbiosis”, or “auxiliary systems to help with the acquisition of knowledge” (76), smells as sweet no matter what the nomenclature. This concept was not firmly defined, perhaps because it refers to a number of technologies rather than a single one, but appears to indicate communication between people facilitated by computers. The term indicates the shaping of the computer to fit human needs, when executed so well that the machine is as close and comfortably fit to human needs as a glove:

” Man-computer symbiosis is a subclass of man-machine systems. There are many man-machine systems. At present, however, there are no man-computer symbioses.” (77).

Licklider’s paper suggests new adaptations of computers very far from any existing I/O or peripherals, or applications then available. He complains that plotting data and drawing up charts is slow, and says that the computer applications ought to be carrying out such things (78). In the interest of helping the scholar or the engineer, he suggests that computers execute graphs and charts, read handwriting (“perhaps on the condition that it be in clear block capitals”), hold interim notes on a clipboard-type display surface, and finally recognize and produce speech. Licklider also requests that the user “should be able to present a function to the computer, in a rough but rapid fashion, by drawing a graph...” (79).

All of these functions and more are of course available in 2012 on any personal computer with an Internet connection. The idea of augmentative or pro-cognitive systems, refers to a desktop equipped with tools for natural language recognition, graphing and charting tools (i.e., software applications), all at the disposal of the engineer. This reach greatly exceeded the grasp of the time. Likewise, when Licklider and his cohort of friends surveyed the actual availability of rapid memory, software to hasten searches by keyword or subject matter, or any machine-readable access to technical data, they found that such resources did not really exist.

Licklider's concerns included the facilitation of scientific research. This is a question of more generic and longer provenance. It had confounded Vannevar Bush at the very start of the Postwar period, when the explosion of scientific knowledge and literature was about to become truly overwhelming. But if both of these people were in the right place, intellectually speaking, only one of them was there at the right time. In 1961 Licklider, the supervisory engineering psychologist at BBN, was recruited to be in charge of a project, sponsored by the Ford Foundation, to question the challenges that the surfeit of knowledge in books posed to libraries, and suggest something that might be done about it. He carried out this project during 1961 and 1962, and his report was published as Libraries of the Future in 1965.

The book does not propose any new technology but examines current developments in computer science broadly understood, with a good deal of consultation with the major computing and AI practitioners of the time (80).

Licklider proposed that the concerns about the perpetual inaccessibility of libraries' materials could be ameliorated with the effective introduction of computers. At the time, computers were particularly involved with paper-based libraries, nor was there any initiative to get them involved. None of the growing surfeit of books and statistics was extant in machine-readable form. If indeed materials were to be put into any sort of computer memory, they would thus be made more accessible. Licklider points to the need for faster access to secondary storage, and the need to get the corpus of academic knowledge into a processible [sic] memory. But at the time the research effort could see nothing that would fit the bill. Magnetic core memory was not that fast, and that was the modal form of memory at the time; the IBM 360 came out with thin film memory a year or two after Licklider did his research. Tape memory was serial and thus harder to traverse. Licklider's report indicates that information retrieval by computer must be relatively rapid to be effective and readily usable, but observes that the technology of his day does not offer "immediate response". Of course it did not: the integrated circuit was not turned in the direction of storage until the late 1960s, even though superior

magnetic core memories were being devised throughout the decade. As Licklider indicated:

"A modern maxim says, people tend to overestimate what can be done in one year and to underestimate what can be done in 5 or 10 years." (81)

Libraries of the Future also suggested software applications, pointing out that natural language, or at least fourth-generation computer languages were needed, and suggesting keyword searches, or "associative chaining as an information retrieval technique" as he called it, and automated card catalogues (82). As a prospective application for the library problem, Licklider and Ed Fredkin of BBN proposed a list processing language to be called "trie memory", which was a storage scheme which would consist of nothing but pointers (Fredkin, 1960 ACM). Is this the first suggestion of hypertext ?

The MIT-BBN group also suggested Computer-aided Collaboration. Noting that sometimes several men [sic] must share dynamic information, Licklider offers up what would today be called a blackboard, in which:

"Some information must be presented simultaneously to all the men, preferably on a common grid, to coordinate their actions.. The information must be posted by a computer" (83).

Finally, several members of the MIT group suggested a scholar-support system which could search for references and otherwise facilitate such work. This vision was realized later in the www, which began of course as a tool to help scientists:

" We wanted to analyze the scholarly process -- reading and studying documents, tracing references, and so on -- and build an interactive man-machine system, or a person-machine system, to facilitate that. Raphael, Kane, Bobrow and I actually wrote a paper about such a thing. About two years ago I was visiting Xerox PARC -- walked in on this group, and they said it was not a put-up job. They were working on exactly such a thing. And they had a copy of the paper, and indeed, there was Bobrow." (84)

J.C.R. Licklider's vision, and basically that of the IPTO during the longer heyday of ARPA, was toward procognitive systems, but it was definitely not an AI vision. He said as much. In fact, Licklider simply concedes the extreme optimistic vision for AI:

“ Man-computer symbiosis is probably not the ultimate paradigm for complex technological systems. It seems entirely possible that in due course, electronic or chemical machines will outdo the human brain in most of the functions we now consider exclusively within its province...” (85)

However, several pages earlier, his statements were far more restrained:

“ However, the AI projects of Newell and Shaw and Simon, various projects Simon and others undertook in psychology, other work at CMU, and the computer language and systems work done by McCarthy and others at his lab at Stanford and by Minsky and the many students who passed through his lab, were all funded by ARPA. But a good number of the ‘procognitive’ innovations were in fact developed, independently, as innovations that made computing far easier. There are in fact several theorem proving, problem solving, chess playing and pattern recognizing programs capable of rivaling human intellectual performance in restricted areas; and NSS’ General Problem Solver may remove some of the restrictions. In short, it seems worthwhile to avoid argument with (other) enthusiasts for artificial intelligence by conceding dominance in the distant future of cerebration to machines alone. There will nevertheless be a fairly long interim during which the main intellectual advances will be made by men and computers working together in intimate association...” (86).

Licklider's incarnation was indeed original, yet the idea itself has an ancestral patina to it. Vannevar Bush's memex is certainly an augmentative device. Moreover, Licklider's contemporary and IPTO beneficiary Douglas Engelbart was struggling at this time to actually realize a computer-aided office at this very time. His paper " A Conceptual Framework for the Augmentation of Man's Intellect"; introduced the concept of hypertext or something extremely close

to it, independently of Fredkin's version (87). A decade later Alan Kaye's dissertation project, the Dynabook, subsidized by Darpa grants to the University of Utah, would embody the same things. The research on the Library of the Future and speculations on procognitive systems did not lead instantly to new technology, but it definitely inspired the applications which were being taken up at the time.

Chapter 4. Hardware, Systems and Applications in the 1960s

Introduction

As we saw in the last chapter, both ARPA and the IC profoundly accelerated extant trends- recursively, we might say. Both the Cold War and the introduction of smaller, faster, more solid-state tech components, were inexorable secular trends which had been moving for at least a decade before the early 1960s. But ARPA fundamentally broke the existing mold. This was true as well of certain aspects of the hardware (memory, storage, IP OP), software, and applications innovations that we will discuss in this chapter. It was also true that the sheer number of computing machines increased greatly, on a linear scale during this time duration. Campbell-Kelly and Aspray tell us that:

"...in 1950 computers were not yet commercially available but by 1960 the nascent computer industry had delivered about 5000 computers in the US and another 1000 or two to the rest of the world... By 1970 the number of computers installed would increase more than 10-fold." (88)

As we saw in Chapter One's discussion on the status quo of AI and the computer, changes made in computing during the 1960s would enhance the scale and scope of continued development of AI itself. The beginning of AI work, in its most pure and theoretical form, predates integrated circuits. The memory requirements and technical form of the earliest AI, that is the Logic Theorist running on punch cards on that SAGE spinoff, the Johnniac, was itself very modest. For the very earliest work, these most rudimentary computing facilities did indeed suffice. But like all babies, this one

grew. The technical sophistication and memory requirements of AI programs increased, even as scientific and commercial computing became less restricted by such antiquities as magnetic drum memories and punch card input. Frequently, as in the case of the contributions that members of Project MAC made to the DEC-PDP, commercial scientific computing and its users were even closely associated.

The capabilities of software programming languages and associated programs improved dramatically in the late 1960s, at the same time as hardware itself took a leap forward. Electronic computing has been a continually dynamic field during the entirety of its life since WWII. But the introduction of the integrated circuit for logic during the course of the early 1960s and the integrated circuit for memory late in the 1960s, helped to make the field far more ubiquitous than it had previously been. The dramatic development of large knowledge bases, the appearance of fielded applications which used a great deal of power, and the beginning of commercial expert systems during 1975 through the early 1980s, were all very much dependent upon these developments in the wider world of computing.

Electronic computing had been initiated by the largest of large users, namely the United States Armed Forces, and indeed its development for roughly the first decade was practically entirely driven by the demand of this one user and its foreign counterparts. All applications at first were tiny barnacles on the side of that grand ship. Commercial demand, weak or nonexistent during the first several years, gradually started for non- military applications through the 1950s. Then after 1960 the military, under the aegis of ARPA, funded projects such as IPTO which were almost always dual use, and commercial use aiming at a wider, cheaper market. The pivotal invention of the integrated circuit was taken up by NASA; and that move, again by a branch of the U.S. military, opened the floodgates to a world in which computers confront us at every turn, often even without our knowledge.

Superior microprocessor capacity- and the existence of the microprocessor itself, was what made possible personal computers and workstations, both of which allowed a far greater commercial

and consumer market than anyone had imagined could exist. The mass commercialization of computers, eventually for both personal use (e.g., consumer chat rooms on casual topics), and as small capital goods, for instance for tiny businesses, in a sense started with the integrated circuit. This development toward computing in every home and office proceeded gradually, and by the early 1980s its progress allowed AI to become oriented to industrial needs as well as to scientific issues. We will discuss this in more detail in this chapter.

2. Memory, Storage, and Input and Output

The usage of computers themselves during the early 1960s was a drawn-out and difficult matter. The tremendous increment in computing speed and efficacy which was realized by the introduction of ICs to computer logic, was realized again in the introduction of IC technology to memory and storage. But this took longer to actually take place, and hence methods which were more archaic, if otherwise adequate, persisted far longer. As developed in primordial form by Charles Babbage, memory is of two forms, the first being the store, or the accumulators which hold intermediate results as it is being calculated, and the second being the mill, or information held in card form in the Analytical Engine and brought forth when requested. Memory of the former type is the only one typically referred to as memory, or internal memory in lingo, while the latter is called storage, or external memory. They are obviously both memory in the sense of being stored information, but the nomenclature has stuck because it is useful for intimidating people with common sense but no expertise in computers.

At the end of the 1950s, dynamic memory, or the accumulators, was often held in the form of flip-flops, electro-magnetic circuits with binary states, or magnetic media of various forms. The method was both insanely archaic to our eyes, and absolutely novel and brilliant at the time:

“ The magnetic media, in several forms such as drums, tapes and disks, used the common technology of ferromagnetic coatings. The coatings were applied selectively to the surface of the medium to represent binary digits.” (89)

Internal memory was often taken care of with a magnetic drum storage, which looked like a giant metallic hair roller. At about two feet long and a foot in diameter (90), it was larger than any desktop computer today. Magnetic cores, one of the engineering triumphs of the early 1950s, were still used for memory, and magnetized recording disks, usually large ones ten inches in diameter, were used for storage of programs. Magnetized media remained an area of manufacture and study, and was subject to many exotic permutations thereof, such as laminated ferrite storage, grid slot ferricle, and woven screen storage (91). Punch cards remained a significant form of storage.

The persistence of the use of magnetic media memory and storage may seem odd. But we should keep in mind that thin film storage (and logic and memory) using semiconductor technology had been under development almost since the digital computer itself. Yet it was still not the obviously superior technology. Its advantages became more and more pronounced during the 1960s, as is evidenced in the great success of the IC firms, especially when these firms began to apply semiconductor technology to memory and storage. Moreover, magnetic media memory and storage are still with us, in the form of floppy diskettes still occasionally used for storage.

Software and Systems

As long as there have been computers, even the experimental ones such as the EDVAC, there have been users. But these users have been mostly indirect; the programmers have typically interpreted their problems into inscrutable programming code. Because of the indirect nature of the interaction with users, and because of the instincts to self-preservation perhaps inevitable in any profession, the tendency toward job-preservation among the programmers has occasionally discouraged advances in programming. Fortunately, this has always been overridden by technological advances. This is evident as early as the appearance of rudimentary programming techniques such as jump instructions and assembler code in the early 1950s. Because assembler code proposed to make programming easier and more 'automatic' than preparing tape or

punch cards directly in machine language, early programming software became known as 'autocode' (92). Even in the 1960s, what we now call programming was sometimes called 'coding', although the term software was in use by 1960 or so (93).

Each of the very limited number of manufacturers of computers, as well as some independent universities, provided their own programming language along with the hardware; Fortran from IBM, Flowmatic and Math-matic from Remington Rand, Madcap from UCLA, and the oddly named Nebula and Mercury from Britain's Ferranti (94). These "autocodes" permitted programming essentially in pseudocode rather than the numbing visage of endless binary numbers. Given the intellectual weight of IBM, it will not seem too surprising that that company produced the programming language which is most often noted. The Fortran Compiler, created in the mid-50s by J. Backus, remained in use in scientific and numerical computation for more than a decade (95).

The Consolidation of Formal Computing Languages and of Programming

Yet a participant with even more clout than IBM was needed to stop the impending problem of too many computer languages, entirely independent of each other. The software equivalent of a tower of Babel was accruing, with the accompanying compulsion that customers in industry and government learn new (computer) languages and conventions, as well as actually rewrite all of the existing programs, every time a new computer was acquired. Like complexity in transistorized computing, this problem was brought about by the success of the medium rather than its weaknesses. As programming became a permanent career rather than a divergence from electrical engineering, common conventions in the linguistics of formal languages became a necessity. Fortran became a de facto scientific standard. Meanwhile, the U.S. government sponsored the creation of a computer language by committee in 1959, declaring that it would virtually refuse to lease or buy any new computer which could not run Cobol- the COmmon Business Oriented Language. Coercive as it was, this gesture worked. Computer manufacturers were practically forced to use this standard language (96). The creation of standard languages, by hook and by crook, helped to bring in the long slow process by which computer science

became more and more about software, rather than solely being about hardware.

Input and Output

Cultivation of the more aesthetically pleasing computer interface has included a small but ergonomic shape and size, fine high-resolution screens, and peripherals of all varieties. There were no such developments circa the end of the 1950s. Since the world population of computers numbered below five digits, the user was supposed to be grateful for any access at all. Such input and output as there was, was as functional and Spartan as a medical laboratory, with dull tan cabinetry, chilly air conditioning to keep the precious componentry cool. Often, the components were still vacuum tubes; there was not a CRT screen in sight. The important thing- apparently the only important thing given how much preparation one had to go through to engage in any form of computing- was that digital computers did indeed work.

Throughout the course of the 1960s, the implementation of the IC accelerated. Thus the way that computers worked, in terms of arithmetic and logic units, was increasingly novel. In terms of storage and memory too, computers often relied on a faster and more intricate magnetic disk. But input and output were often carried out in nearly the way that the user's grandfather might have done it, using the IBM-type punched cards settled upon in 1928. Generally, the input method was the same as the output method. Punch cards or paper tape, prepared at keypunch machines or other special typewriters, were fed to the computer by high-speed special feeders (IBM had manufactured these since the 1920s). The machine would respond by spitting out tape or punch cards as well, and the latter were fed to 'electrical typing machines" or gigantic high-speed printers which typed out results for human scrutiny (97).

Around 1960, the preferred input began to shift from punch card machine production to magnetic media for input and external memory (as mentioned earlier). IBM introduced a line of dumb but not mute machines which transferred data from cards or paper tape onto much faster magnetic tape reels, and then read the outputted results from the magnetic tape to a high-speed printer (98). But the

magnetic tape was much more expensive, and punch cards persisted. They were to be found used as input media in countless settings decades later (99).

Now that visually pleasing computer peripherals are taken for granted, it is difficult to imagine machines that don't have them at all. But as late as the middle of the 1960s, even IPTO administrator J.C.R. Licklider referred to a computer screen as an oscilloscope (100). As we have noted, both input and output were traditionally conducted with cards rather than with any more immediate or appealing window into the activity of the machine. The computers of the 1950s talked to the user with various tiny tubes and bulbs, and were talked to with toggle switches as well as by cards. But by the early part of the 1960s, the Flexowriter was coupled with early screen displays. Visual read-out devices featured alphanumeric characters in rows, glowed a bright and ominous green, and emitted perceptible electrical energy. Early screens, or digital information display systems, such as that made by Raytheon, were housed with the typewriter units in cabinets roughly the size of a refrigerator (101). The visual display we refer to was only for output, displaying data from its buffer (4096 nine-bit words), but screens that were used for as devices for online programming editing were also being used, in conjunction with a typewriter unit called a Flexowriter (102). This was used, for instance, in the TX-0 (103).

As of the late 1960s, though, using a CRT screen remained the exception rather than the rule. Newell, Shaw, and Simon worked with the Johnniac for a decade, and commercially profitable "computer trainer" machines offered toggles and push-button switch settings (104).

The near-absence of CRT screens, and the (decreasing) rarity of Flexowriter devices, underlines the nature of man-machine interaction at this time. The relationship between the user who wrote programs and the human system operators who were empowered to actually operate the machine has been characterized as one of acolyte and priest (105). Given the skeleton of the circumstances, the description seems apt, but the sheer ardor of the computing process must have imposed some concomitant rigidities.

Beginning in the early 1960s, the mainframe computer manufacturers generally declined to either implement timesharing and more diverse I/O methods, or to produce smaller minicomputers for research and later for hobbyists. The latter niche was filled by DEC, and later spawned the PC. Certainly, during the period we are referring to, we must say that reluctance to implement better “man-machine interaction” was a deliberate decision. However, there was a widely stated philosophy behind it. It is intriguing to see that there was widespread belief that since humans could never keep up with the rapid calculations of computers, direct (online) input by typing was of no use. A.D. Booth, an extreme Luddite for a computer scientist, is suspicious as to the need for actually truncating input and output by allowing direct access with a keyboard:

“ It may be inquired whether or not direct communication with the machine from a keyboard and direct output from the machine via a typewriter are used. The answer in both is yes, although for input direct human intervention in machine activities is not favored, as this is a slow operation and one only to be used in engineering tests, or in altering a few instructions in a programme which has been previously inserted by an automatic reader.” (106)

A technical solution to this problem was quite clear. So the deliberate decision not to design computers with buffer zones for programmer editing of IP was clearly about an issue other than iron technical barriers. Instead, the decision concerned the anticipation of what computers could be. One reason not to introduce buffer zones, systems interrupts, and other modifications needed for multiprogramming and timesharing was the human systems operators who wished to hold onto their jobs. John McCarthy stated that the various stalling methods put forth by interested parties at MIT were “analogous to trying to establish the need for steam shovels by market surveys among ditch diggers” (107).

McCarthy, and a growing cohort of computer scientists and students mainly concerned with software, wanted and needed to have better access. They wanted to be able to debug their programs online, by reading the input through CRT screens and punching in corrections

through the Flexowriter, rather than having to wait up to three days to find out the error of their ways. The need for negative feedback, so to speak, was more immediate, and the lack of access to computer time greatly impeded the programming work by McCarthy and others at MIT. This sentiment resulted in the development of timesharing, which was originated by programmers at the very end of the 1950s, and multiprogramming, implemented in the SAGE project and reinvented again late in the 1950s. In Timesharing, programmers occupy multiple terminals with CRT screens and flexowriters, and their programs share one central CPU unit. In multiprogramming, the CPU actually works in batch mode, but because of the storage of inputted programs in buffer zones, the programmers perceive the situation as a timeshared one.

Computers for Research

Complex Information Processing, as Newell and Simon had initially referred to AI, proceeded at the Carnegie Institute because of the special dispensation arranged between Allen Newell and the Rand Corporation. This allowed Newell, Shaw, and Simon hundreds of hours of remote computing time over many years. NSS, John McCarthy, Minsky, and their students were habitually privileged to be permitted to beta-test IBM machines. McCarthy, and others, were also given time at the New England Computation Center, to work at IBM for a summer on the new computers. This alone was not adequate to instigate an intellectual movement of any magnitude. They needed funds, which ARPA provided, and which we will discuss this in a moment. But they also needed certain sorts of computers, namely minicomputers which had features such as offline editing of one's programs using a teletype machine and CRT screen, and timesharing. Such machines were first developed late in the 1950s, and tremendously facilitated AI and computer science research.

As we have seen, the desiderata, and increasingly the possibility of I/O for computing, was found in opportunity to edit work online with a keyboard and a CRT screen, rather than submitting punch cards; to work in timeshared mode; and to receive programming results faster. The latter was in the control of hardware designers, who were becoming entranced with the dream of faster computers with

tinier components. Of the two former hopes, the first required money, which AI would get hold of in good quantities starting in 1962, and the second required both work in programming and help from commercial computer companies (we will discuss timesharing in a moment). All of these desiderata were also helped by the appearance of computers made by and for computer scientists, namely those introduced by the Digital Equipment Corporation (DEC).

One of the great contributions of the SAGE system was the CRT screen. The FSQ/AN7 SAGE computers were locked away in secret bunkers, as befitted a project concerned with control of nuclear weaponry. However, some of the innovations introduced for the SAGE project appeared at Lincoln Labs in the mid-1950s. One such innovation was the TX-0 ('tixo'), with a CRT screen and keyboard I/O design. Access to this computing machine was highly limited, but even so the item drew students like a doughnut draws ants. In 1959, Lincoln Labs gave MIT proper a TX-0 (108). Its arrival instigated immense interest among the tiny band of 'computer slugs', or computer bums as they were often called at the time. They wanted time because of interest in the highly interactive computing the TX-0 offered, and because of a course given by McCarthy which introduced LISP (109). At first the machine was neither batch-programmed nor time-shared. Instead, but users were given slots of exclusive CPU time. For students, this time was often at night; hence the birth of nocturnal programmers. The students also created the software with which the machine had not arrived.

Imitation remains the highest form of flattery. The formation of a company that produced a functional equivalent to the TX-0 followed extremely closely upon the introduction of the TX-0 to Lincoln Labs itself. In 1957, two Lincoln Labs engineers left MIT to form a very early start-up company in a Boston exurb, manufacturing machines that had the TX-0's best features. DEC founders Kenneth Olsen and Harlan Anderson dreamed of computers which were small and cheap, at least small and cheap relative to those then available from IBM. They saw Big Blue as the great blue nemesis, despite the fact that IBM had historically been responsible for the lion's share of commercial computing. Olsen and Anderson competed in many ways, not the least of which was price. The PDP-1, DEC's first

machine, sold for \$120,000 (110). These computers had the I/O features of the TX-O, and were transistorized as well, with the attendant advantages of speed and reliability. The earliest PDP, or Programmed Data Processor, computers went to BBN and the RLE at MIT in 1961. After that by all of the major AI centers (and other university and scientific programming facilities) got hold of their own. DEC ardently sought ways to improved their products, and hired MIT's hackers away in order to find out about their system improvements. The usage of the PDP at MIT resulted in a virtuous cycle as well. In the opposite direction, improvements were brought to MIT computer science by the DEC machines, even while MIT students produced operating systems, computer games, ersatz word processing software, and other utilities (111).

Moreover, the influx of ARPA funds brought with it a dramatic improvement in access. Project MAC, an acronym that meant multiple access computer, or machine-aided cognition, or man and computer, or finally, "More Assets for Cambridge", delivered everything that its odd name promised. By 1963, the Project Mac initiative had indeed greatly increased facilities for students' (and others) computing. The RLE students, along with Minsky and Seymour Papert and the earliest AI graduate students, moved into more spacious quarters at Tech Square (112). DEC introduced further versions of the PDP, and research-oriented computing in general was greatly improved as DEC itself split into two. The succeeding company, Data General, was joined by a number of other minicomputer producers. In the 1970s, the minicomputer and Workstation niche would greatly improve in price competitiveness, and ergonomics, as we will see in Chapter 10.

The Improvement of Timesharing and the Computer Utility

However much the term has been debased by advertisers and public relations flacks, "visionary" is the appropriate label for Licklider and his cohort. It would even appear that they did the concept of visionary one better, as he and his colleagues began to bring to life the things they foresaw. The ideas of procognitive systems, eventually manifest in the computer mouse, file systems, word processing editors, and bit-mapped screens, did indeed get started during this period. The ARPAnet, unequivocally conceived

and brought into being by the IPTO people, with some involvement by AI, was also initiated during the 1960s (113).

All of these things took decades to get enroute, but timesharing was put firmly underway by the early 1960s. It was this improvement in computing systems which had the most direct impact on AI and most participation by AI figures. As we saw in the previous volume, the essential idea of timesharing from the perspective of the processing unit is the system interrupt, which stops one program in order to implement a second (or third, etc.) computer job. The first implementation of this idea had been in the Atlas computer, designed in Britain in the mid-1950s in cooperation between the University of Manchester and Ferranti Ltd. (114). The Atlas, in which a system interrupt was essential to the proper functioning of the virtual memory, was intrinsically timeshared. The Atlas' engineers originally included timeshared terminals in their designs, but this design feature was removed from the blueprints as being too expensive (115).

When design is absent, happenstance may prove serendipitous as well. The system interrupt had not been introduced as a way to alter the nature of human-computer interaction, but this feature did just that. The essential idea of timesharing from the perspective of the user is many people preparing and editing their programs simultaneously, at consoles which combine teletype and CRT screen units. But the demand for timesharing did not grow only from the side of corporate computing: timesharing was a populist movement, borne by self-organization among programmers, as well. Late in the 1950s, the technical model of timesharing was independently conceived again by others. British engineer Christopher Strachey conceived of the concept of timeshared programming through the addition of several individual operating and card-reading consoles adjacent to the mainframe computer (116). In addition to Strachey and the Atlas designers, John McCarthy also apparently invented timesharing, including most of the modern I/O and programming features, on his own (117).

Developments through the 1960s

IBM has come under wide criticism in the computer histories for being slow to implement timesharing. The hackers and programmers may rail, but at least at first, the firm was observing the reality of circumstances. To put it bluntly, systems advances such as timesharing were somewhat gratuitous and hardware ruled. At least it ruled until the IC was widely implemented in the early 1970s. The big computer manufacturers may be faulted for not energetically implementing timesharing and superior I/O for the scientific and hobbyist markets late in the 1960s, but in truth they did encase new IC arithmetic and logic units behind steel cabinets and market them quite rapidly. (Sometimes even too rapidly, as in the case of the IBM 360, which introduced a close relative of IC logic, as well as upward migration paths and massive system improvements in the mid-1960s and initially lost money).

It may well be argued that IBM and other major manufacturers began to lose their technical lead and vitality during the 1960s, while maintaining their reliability. Certainly DEC replaced IBM in the role of major supporter of scientific computing, or at least computing for AI. This role is important to our story, even to its highly intellectual components, because power and speed and memory were necessary for research computing. DEC appears more prominent, but remained minor in the larger role of computing companies. The major companies seem to become somewhat faceless, but their role in the sweeping introduction of access to computing into the university does not diminish. This is simply because they were the grand actors in increasing production, implementing cheaper and faster logic and memory and storage, manufacturing Flexowriters and consoles to read at them, and dropping costs, and thus all the while turning the trickle of output of computers into a stream and then a flood. Through the course of the 1960s, visible change in implementation of the computers through businesses and government functions took place as the supply increased immensely. The manufacturers became bigger companies: IBM moved into the Fortune 10 during the 1960s, and gathered more than 50% of the market in computing itself (118).

Notwithstanding this redundant and exaggerated triumph in commercial computing, IBM's role in AI receded. IBM did not adopt timesharing until 1964 or so. Programmers who wanted this feature

had to modify the equipment by themselves. The modifications, which were widespread, were also illicit, since IBM's computer manufacturing and sales and service functions were separate, by court anti-trust orders (i.e., the 1956 Consent Decree in which the Department of Justice ordered IBM separated to weaken its monopoly power). But the idea was sufficiently technically possible and so helpful for programmers that both the supply and demand sides helped to make it happen. Originally, pleas to IBM resulted in hardware modifications to the IBM 7090, which MIT received around 1960 (119). Then, the newly formed Digital Equipment Corporation was persuaded to modify their computers to allow a higher level of interactivity. DEC hired several MIT electrical engineering graduates who were familiar with the equipment and the need for greater flexibility, and collaboration resulted in the new PDP-1. In the next round of modifications to the DEC line, McCarthy (now at Stanford), succeeded in designing a time sharing system which used display terminals rather than punch card output. Soon the DEC machines, rather than the IBM equipment usually leased at easy rates to universities, became the machine of choice for AI laboratories (120).

McCarthy advocated timesharing, in speech and writing. His proposal that MIT adopt and champion the innovation was taken seriously. MIT called a committee to study the topic. The Long Range Computer Study Group consisted of electrical engineers (including Jack Dennis, McCarthy himself, and Marvin Minsky). The panel concluded that time-sharing was a good thing and that there ought to be more of it on the campus (121). The work on timesharing may not have been taken as seriously as McCarthy would have liked: this is considered one of the reasons why McCarthy accepted Stanford's offer in 1960, and did not wait to hear what the Long Range committee had to say (122). MIT implemented a timeshared 7090 in the RLE, to the immense benefit of the early students and hackers who needed computing facilities. Next, the MIT Computing Center, which was the service center that took in computer jobs for other institutions as well as MIT, set up CTSS, another such system (123).

But it was ARPA's money that enabled MIT to put its money where its mouth was. ARPA greatly encouraged timesharing. Licklider favored institutions which implemented it, and gave MIT a further

three million dollar grant to develop Project MAC. The Institute set up a summer school course in 1963, intended to encourage corporate as well as scientific users. The same year Project Mac opened its first large timesharing system, which allowed up to 30 users at a time to work remotely. (The applications included scientific programs as well as document editing, paving the way for word processing systems). In part at least because of the loss of both the MIT contract for providing terminals for the Project Mac timesharing project and for the successor, a much larger project known as Multics, IBM implemented a timeshared operating system for a later release of its System 360 (124).

Repeated official forays into timesharing at the Carnegie Institute of Technology, Stanford, Dartmouth, SRI and other computing institutions throughout the 1960s made the technology workable. Basically, the continuation of timesharing at MIT consisted of subsequent projects, which allowed more and more users. By 1965 Project Mac was ready but overloaded, so a new and bigger project was started- MULTICS, for use by several hundred people. IBM was turned down as hardware contractor in favor of General Electric:

“ Multics was to be the most ambitious timesharing system yet costing up to 7 million, it would support up to 1000 terminals, w 300 in use at any one... Bell Labs became the software contractor because the company was rich in computing talent but was not permitted, as a government regulated monopoly, to operate as an independent computer services company.” (125)

Multics would be a classic dual-use technology project, providing the organizational and technological springboard for Unix operating system and the C++ programming language, as well as a secure operating system for military usage by Honeywell (126).

IBM moves into Timesharing

Receiving negative feedback can be productive, if the response is constructive change:

“ Having been pushed aside by Project MAC, IBM’s top management realized that the company could bury its head in the sand no longer and would have to get onto the time sharing scene. In August 1966 it announced a new member of the System 360 family, the time sharing model 67. By this time most other

computer manufacturers had also tentatively begun to offer their first timeshared computer systems" (127).

In addition, in 1966 IBM began introducing a new System 360 series computer for timesharing service in more than two dozen major American cities. IBM has never met an anti-trust lawyer who liked it, and blanketing the country [sic] with a new product or service spelled trouble in the form of anti-monopoly charges. Putatively, the company was dumping the "Call 360 Basic" time-share service below cost to drive out the competition (128).

Notwithstanding the sheer bulk of Big Blue, two dozen other commercial timesharing computer systems appeared during the late 1960s. The computer utility combined with commercial timesharing did not appear immediately as dreamed of, and did not succeed commercially (129). This would have to wait for minicomputers and the personal computer with networked applications, roughly two decades later.

Unofficial forays into the same topic by hackers at MIT and at Bell Laboratories, which was the software contractor for Multics, engendered a better software system, known as the ITSS (Incompatible Time Sharing system). Taken up by researchers at Bell Labs, it was transmuted and eventually became the much-derided UNIX operating system, used in scientific and university settings. CTSS was the official timesharing system, which worked parallel to Multics. ITSS later surpassed the official CTSS (Corbato) and Multics (other official system), in that it allows both more than one user at a time, but also one user to run more than one program at one time; also online editing of the program. The hackers did not succeed in establishing their system as the timesharing OpsSys that would run on all Dec PDP 10s (130). Dartmouth developed a timeshared version of BASIC, which was introduced to 'regular' non-computing students in 1964 (131).

The Concept of the Computer Utility

Notwithstanding the lack of immediate success of the computer utility and commercial timesharing, establishing a dream publicly is in itself efficacious. The idea of the computer utility, a metaphor

comparing generally available computing power to electrical power, inspired all sorts of ideas for applications which were adequately far beyond the present technology that unlike 'procognitive systems' they were not particularly useful. Martin Greenberger, a professor at the MIT Sloan School of Management, suggested a thousand prosaic uses for computing. These include universal credit cards, which he refers to as "money keys", online access to financial services such as insurance, computerized clearing of market transactions, and "automatic analysis of information", such as the data taken in at supermarket registers (132). The idea of the computer utility, reinforced by glowing journalistic references, helped to encourage the plunge of big businesses into larger applications, and spurred an industry of commercial timesharing in a number of U.S. and European cities throughout the 1960s.

Throughout the course of the 1960s, visible change in implementation of computers through businesses and government functions took place as the supply increased immensely. The other protagonist in the sheer juggernaut of faster computing was the torrential flood of integrated circuits being produced. As we will see in the next three chapters concerning the growth of the field in the 1960s, hardware improvements and progress in AI programs were a dual relay race, without which the constantly receding goal could not proceed.

Part III. The Belle Epoque of the 1960s

This early period of AI was clearly an early productive phase of a scientific research project- during which the fundamental approach is well-articulated, and there is a good deal of 'low fruit' because the field itself is novel. This intense but rewarding work remained so throughout this time period, at all three of the major founding institutions as well as at several other centers founded during this period. Increasingly, the universities had the infrastructure, both physical and intellectual, to pursue research rather than simply window shop and wish they could buy. Increasingly they also had something to fall back on, in terms of knowing how to do things, such as depict images in bit-mapping, design a program to play a simple game, model the memorization of symbols or sounds, grasp

objects in the physical world or 'move' them in a computational one. As ARPA's Information Processing Techniques Office finally began to grant AI sufficient funds to purchase computers and give students grants, the number of students increased from a few to dozens. In its entirety, the latter part of the 1960s and most of the 1970s formed a veritable 'Belle Epoque' for Classical Artificial Intelligence.

Is it legitimate to see or depict this work as being of one piece? With caveats, the author believes so. Surely at the beginning of this time period, the tactic is highly legitimate. The structure, small size, and intellectual intensity of these environments meant that people worked closely together and affected each other. Given the steady and later rapid growth of AI as a research field, it is true that by a certain point this was no longer a solid or unitary body of work. But certainly early on there was such a unitary body, or rather the obvious three.

First of all, complex information processing, renamed AI, started at the Rand Corporation and a shade later, at the Carnegie Institute of Technology in the mid-1950s, as was discussed in the previous volume (133). AI of the perceptrons and neural modeling sort was underway at MIT throughout the 1950s, but AI following the LISP language and the knowledge engineering approach only really commenced around 1958. Starting in 1962, MIT received monies to be spent on AI, in the form of Project MAC. In 1964 the computer science division was created, inside the department of electrical engineering at Stanford. These places were the central scenes of the action through at least 1970. Thus through the mid-1970s we can segment the field into three centers, with SRI closely associated with Stanford.

However, the nature of university education in the United States in the early and mid-1960s was generally expansionary, as was the Cold War's research arm. AI benefitted a great deal from this wave, and this meant the proliferation of sites where AI was carried out. AI study also began at the University of Utah (by Evans and Sutherland of MIT), at SRI (initially on a small scale, preceding Stanford by several years), at Yale, and at the University of Ohio, during this time period. Even by the 1973 International Joint Conference on AI,

held at Stanford, the participants are from institutions all over the United States- College Park, Maryland, the University of Ohio; U.C. Santa Cruz- as well as from Edinburgh, Scotland, Vancouver, British Columbia.

The High Road And The Low Road

The “high road” versus “low road” distinction offered by Edward Feigenbaum during the mid-1960s provides meaningful heuristic demarcations as to the different areas of research (134).

Feigenbaum discerns these two distinct paths regarding research on generality. The first sought a universal route which would be ostensibly applicable to as many circumstances as possible. The best example of approaches on the high road was, naturally, Newell and Simon’s General Problem Solver: Intelligence as a general capacity- at least at its essence- was also a rudimentary philosophical approach.

This search for the ‘high road’ to universal heuristics for intelligence dominated the early 1960s:

“ By 1965, the search for a general system for developing machine intelligence had become one of the most pressing issues in AI. The area of problem-solving taken in its entirety since the middle 1950s constituted the AI search for generality... Researchers sought a formal system so general that all problems however represented internally, could be translated into the form of proving a theorem in this system yet so specific that general proof methods could be developed for it... In spite of a decade of effort in this area, Feigenbaum could comment in 1968 that ‘we lack a good understanding yet of this problem of generality and representation’.” (135)

The high road in its core differed from its opposite number in the form of the “low road”:

“ Researchers on the low road eschewed a general problem solving system in favor of exploring specific complex domains and tasks in order to (1) test whether ideas methods developed so far in AI could

be used in significant problems and (2) identify new issues for AI basic work.” (136)

Such distinct paths as inquiry into vision or movement as such (low-road), versus general problem-solving techniques is not as cut and dry a distinction as it might appear to be. It does not mean that the fruitful research concerning a particular mode of cognition will have no further implications. A great many actual applications prompted by the research of Feigenbaum himself appear to have derived from a low-road inquiry into a precise cognitive activity. Moreover, a low-road philosophy at MIT engendered more general knowledge representation formalisms in the case of frames, and Minsky’s agency or society of mind theory of intelligence, in the early 1970s. To employ and extend a familiar axiom, the road not taken can be reached later.

The high road approach definitely undertakes a cognitive psychology auxiliary with it at the start. The low road approach, as we saw a few paragraphs earlier, resulted in significant theories of cognition in the form of agency and society of mind (?multiple intelligences ?) theories.

Alternatively, “ The third approach the one we call AI was an attempt to build intelligent machines without any prejudice toward making the system simple, biological, or humanoid” (137). This AI as engineered artifacts approach takes on lighter philosophical assumptions. Marvin Minsky had begun writing a ‘state of the art’ paper as early as the Dartmouth Conference. In 1962, it was finally in some state of completion, and it appears in computers and thought as a review of current work. In “Steps Toward Artificial Intelligence”, Minsky indicates that:

“ There is, of course, no generally accepted theory of intelligence. The analysis is our own and may be controversial...

it is convenient to divide the problems into five main areas:

search,

pattern recognition

learning

planning

and induction.” (138)

Hard AI and Soft AI

A final distinction that became more salient as the range of research undertaken under the auspices of AI increased during this time period is that of hard versus soft AI. The former indicates a proposal that AI can embody human intelligence- that it can recreate something that is on a par with that of human beings. Soft AI takes on much weaker claims, specifically only that AI is an engineered form of intelligence, or different intelligences. It does not claim efforts to humanoid intelligence, and instead tries to create engineered artifacts, which may be re-engineered or reverse engineered intelligence. We will encounter these distinctions again and again, because they figure heavily in the philosophical debates that surrounded the field later in the 1970s, and continuing into the neuro-philosophy issues of the 1990s.

Chapter 5. MIT: Work in AI in the Early and Mid-1960s

The Origins of AI at the Massachusetts Institute of Technology

As we have seen, in the early 1960s, research funding for practically every scientific field which could have weapons ramifications, and especially for computing boomed. In the midst of this general U.S. trend, MIT's long streak of good fortune was boosted even higher. But this was nothing new for MIT.

Cambridge was the fermentative original location for military-industrial science in the 1940s, and by the mid-1950s the actors, stage sets, and institutions for the next act were in place. Despite Harvard's initial advantage in computing, it was eclipsed as a center of computing hardware during the Second World War, and as the center for software development later on. In contrast, the Institute has been central to computing history for at least half a century (139).

MIT was founded in 1862, and began to admit women as early as the 1870s (140). For the better part of a century it taught engineering from a relatively limited perspective, as simply the reengineering and incremental improvement of existing machines. During the Second World War, its identity changed. The National

Defense Research Council, headed by MIT professor Vannevar Bush, heaped funds upon MIT and particularly upon the Research Laboratory of Electronics. MIT was “the single largest wartime research and development contractor” (141). Stuart Leslie states that:

“...at the end of World War Two, MIT was the nation’s largest non-industrial defense contractor, with 75 separate contracts worth \$117 million, far ahead of second-place Caltech (\$83 million) and third-place Harvard (\$31 million)” (142).

This did not change but was rather enhanced after the war, as sponsored research covered 80% of MIT’s operating budget during the 1950s and 1960s (143). Following the end of the war, the research “troops” were not demobilized, but were redeployed on campus, to the Lincoln Laboratories and to greatly strengthened MIT departments. Some laboratories were simply renamed: the Charles Stark Draper Laboratory was born as the Instrumentation Laboratory during the war, as was the Research Laboratory of Electronics, which was renamed the Radiation Laboratory, or “Rad Lab”. The previously unheard-of memory and storage requirements of the early warning system for a Soviet ground attack, Project SAGE, brought previously unheard-of levels of financing to computer hardware. The tremendous largesse also resulted in the foundation of “hard”, that is highly quantitative, social science departments such as linguistics and psychology, and in the bolstering of the Institute’s pure science facilities.

Through the 1950s and beyond, the Institute became even more intensively intermingled with military contracting: by the beginning of the 1960s:

“its contracts with DoD totaled \$47 million, plus additional obligations of some \$80 million to its federal contract research centers, Lincoln and Instrumentation Labs... prime military contracts for 1969 topped 100 million...” (144).

This figure may be accurate even without the inclusion of MITRE, the Air Force contractee which had been spun off from MIT in 1958 (145). Leslie points out that MIT became so industrialized, and so

devoted to constructing weapons, that its identity seemed itself confused:

“ Sizing up MIT in 1962 from his perspective as the Director of the Oak Ridge National Laboratory, physicist Alvin Weinberg, who coined the term ‘big science’, quipped that it was becoming increasingly hard to tell whether the Massachusetts Institute of Technology is a university with many government research laboratories appended to it, or a cluster of government research labs with a very good educational institution attached to it.” With nearly 100 million dollars in annual government-sponsored research contracts by the early 1960s (a figure that would almost double by the end of the decade), science and engineering at MIT had become big business.” (146).

As at Stanford, although not at CMU, digital computing research started as a service department, rather than as an object of academic study itself. MIT’s Computation Center was founded in 1956, with an IBM 704 computer and money from the Office of Naval Research. It was intended as a service bureau for thirty universities and research centers in the Northeast. Scientists would present programmers with technical problems requiring punishingly long calculations, and the answers would be provided. The problems were of a wide variety, including:

“...fallout radioactivity in rainwater, a dynamic model of competition between two firms, a heuristic strategy for computer game playing, United Nations office operation, shop motions in irregular waves...” (147).

In its first few years, the Computation Center’s major concern was with the improvement of the computing infrastructure- which concern led in turn to McCarthy et al.’s timesharing (148). Whatever its necessity in the context of its times, this did not allow for hacking or student usage of computers.

MIT’s AI community reached well beyond the university itself. We have mentioned several laboratories with computing facilities. A description of the community is also incomplete without BBN, a spinoff of several MIT engineers who built a consulting business. For

most of the half-century duration of the Cold War, Bolt, Beranek and Newman was closely aligned with Harvard and MIT faculty members and graduate students in computing sciences. MIT professor and IPTO director Robert Kahn called it “the cognac of the research business” (149). A number of the AI researchers mentioned in this chapter and the ones to follow were employed at BBN, which used AFOSR funds to subsidize their graduate studies (150).

The Establishment of AI Per Seat at MIT

Artificial Intelligence at MIT specifically, as much as computing in general, developed at first from the bounty spilling from this set of historical circumstances. Marvin Minsky and John McCarthy joined the Mathematics department at MIT in 1958. They established the Artificial Intelligence Project at the start of the 1958-1959 school year. In 1960, MIT dean, and later president, Jerome Wiesner encountered McCarthy and Minsky in a hallway. He asked them what sort of facilities they needed. Their requests were minimal: an office and keypunch and two programmers. Wiesner instead he lent them the labor of six ‘redundant’ graduate students from the Research Laboratory of Electronics. The RLE students were supported by JSEP block grant, and thus could be put at McCarthy and Minsky’s disposal (151). IBM supplied the equipment. The Office of Naval Research, under Martin Denicoff, also acted as a sort of ARPA before ARPA, providing other funds which bolstered the volume of research that had been done in robotics prior to the existence of the IPTO (152).

AI Research in the Early and Mid-1960s

The Birth of Hacking, and Its Rewards

hack: “... 2. n. An incredibly good, and perhaps very time-consuming piece of work that produces exactly what is needed”.. 6. vi. To interact w a computer in a playful and exploratory rather than goal-directed way... 9 [MIT] v. To explore the basements, roof ledges, and steam tunnels of a large institutional building to the dismay of Physical Plant workers and (since this is usually performed at an educational institution) the Campus Police...” Microsoft Press Computer Dictionary. Second Edition. 1996.

Practically all cultures- nations, ethnic groups, artistic genres such as theater and painting- have both a ‘high’ tradition and a ‘low’ one.

For instance, Latin contrasts strongly with the vernacular languages of Early Modern Europe. Strindberg and Ibsen have moved many people, but 'Seinfeld' and 'The X-Files' drew larger audiences. And so it is with artificial intelligence. All of the major figures who founded laboratories and research programs have played the role of the leader of high culture. But all have also presided over the folk tradition of AI in their laboratories. The functional equivalent of low culture in AI is hacking. Like low culture itself, it is viscerally appealing, useful, interesting and not at all profound (that is, it does not directly address 'big questions'). It is often small in scale (at least initially), or calls upon few resources to create it, and is often made by people with relatively less formal education than the adherents of high culture. As the Rolling Stones said, 'I know it's only rock and roll, but I like it'.

Hacking, the folk culture of AI, consisted of work done largely by undergraduates and others not particularly cognizant of or concerned with their place in the graduate school-to-professor pecking order, and largely outside the auspices of academic progress. Some people at MIT, and their work, have moved along the regular academic track. Gerald Sussman, who began with vision research in the mid-1960s, for instance, has become a renowned professor. But many hackers seem to have been drop-outs. Likewise, much of the work done in hacking has concerned the emulation of intelligence- in vision, robotics, ergonomic utilities which facilitated the cumbersome features of computing and applications. But this work has generally been not much concerned with theories of the mind as so much of AI has been. It has not had an academic axe to grind, in either a positive or a negative sense of this phrase. Hacking is sometimes defined as learning about computing by trial and error, or usage of computers in a random way or for fun, or for enjoyable projects rather than some scientific purpose. The same sort of tinkering can take place upon other objects of study, in areas other than computing or AI. Computer hacking thus has a pre-history, just as formal AI as the physical symbol system or different forms of representation of memory has a pre-history.

We have established that AI as high culture was initiated, with a very few people such as Newell, Shaw and Simon at Rand and

Carnegie Tech, in the 1950s. But there was plenty of pre-AI, in the creation of the Perceptron, among the Cybernetics group, in the early work of the cognitive scientists, in the theories of automata and self-organizing systems by McCarthy, Minsky, Selfridge and Von Neumann. This is the prehistory of formal AI. There is also a pre-history to the other sort of informal AI, better known as hacking. There is a significant hobbyist tradition in all electrical-related areas- radio sets, electrical wiring, model railroads, car mechanics, self-guided mobile planes, and the like. Much of this continued in force through the latter part of the Twentieth century, but was gradually eroded by the increasing popularity of computing, and its ready availability after 1985 or so. Popular Mechanics, a serial devoted to electrical and mechanical projects for home-garage hobbyists, enjoyed immense popularity in the United States during the PostWar decades. Electro-mechanical engineering hobbies, epitomized by model train clubs- MIT's being the most famous (153)- and other Popular Mechanics-types of tinkering, absorbed people who might otherwise, or later, become entirely entranced with computers.

The Birth of the Concept of Hacking

“ ‘Hacking’ was used by many MIT students to describe any activity undertaken to avoid studying- this could include goofing off, playing bridge, talking to friends, or going out... In the middle to late fifties, additional meanings for the word hack were developed by members of the Tech Model railroad club, including an article or project without constructive end or an unusual and original solution to a problem such as inventing a new circuit for a switching system.. The word hack found its way into common usage outside MIT with the advent of computing hacking in the early sixties.” (154)

In Nightwork, MIT Institute Historian T.F. Peterson establishes the term “hacking” as a product of MIT in the 1950s. (Peterson, incidentally, may himself be a hack rather than an actual person). This Ur-hacking was first turned away from electrical hobbies and toward computing in the tinkering of a dozen or so young students at MIT's Model Railroad Club (TMRC). In the absence of computer facilities accessible to undergraduate students, this opportunity evolved into elaborate programmable switching programs. Students

were given complete control and as much time as they wanted at the Club's elaborate train system (155). Potential students lacked any opportunity to work directly with computers, and thus resorted to pursuits which could absorb as much technical ingenuity as they wished to throw at them. The first computer hackers were apparently recruited to, or drawn to by word of mouth, from this cohort, at the very end of the 1950s. Minsky built alliances with the students in the popular Model Railroad Club, where "indigenous computing" was already in practice (156). John McCarthy taught one of the first college courses in computing, and arranged to allow undergraduates computer time to punch and run their own code for the course.

Likewise, at Carnegie Tech, students obtained usage of machines through graduate professors. Newell and Simon and colleagues achieved early results, it seems, because they offered early computer access for their students. Ed Feigenbaum, for instance, took a course in IBM 701 programming with Simon in 1956 (157). Starting in 1959, McCarthy taught a course which required programming on the IBM 704, and offered CPU time for students. Minsky and his EE colleague Jack Dennis began to cultivate the friendship of a number of the students who had moved from MIT's Model Railroad Club to the computer.

This was quite a revolutionary concept at the time. The status of computing, at MIT and elsewhere, circa the end of the 1950s, was certainly better than it had ever been technically. But computers remained infinitely inaccessible to the average interested party. Until Digital Equipment Corporation was founded in 1957, IBM's rivals were considered paltry. The industry was characterized as "IBM and the Seven Dwarves", and thus the practitioners of AI and every other usage of computers had to accept the computing that they were given. The machinery for computing was, in academic settings, typically an IBM 704. The machine itself placed stringent limits on the nature of access to computing time. The IBM 704 required two larger rooms and constant monitoring in case the special air conditioning broke- which happened often. Only people with some official usage, typically with some connection to Lincoln Laboratory or the RLE, were allowed to hand over punch cards to the machines systems operator.

This effectively ruled out opportunities for curious undergraduates, at MIT and elsewhere (158). Regular people were no more allowed to work the controls of a mainframe computer than they are permitted to operate the controls of a nuclear power plant today. This applied even to MIT students. “Friendliness” to the end user may seem like a commonsensical notion to the reader early in the Twenty-first century, but in the middle of the Twentieth, it was by no means obvious. Because of the sensitivity of the work done by these machines, and because of their very proneness to error and breakdown, highly restricted access and layers of guards were the rule. We should also consider that the management philosophies of the time were still not terribly far removed from the Pre-WWI philosophy of scientific management, which purported to shield valuable capital machinery from worker ‘ineptitude’ or sabotage.

Furthermore, there was also very little pressure from users to help initiate further innovations in computing. The tiny number of people in the field did not exert a great deal of pressure on the demand side of the market. The miserly supply of computer time even for professors was commensurate to the generally limited demand for the machines. The stringent limitations of any individual’s time with the computer were due in part to the larger lack of active demand for computer time in all but the biggest corporations and most elite university-military settings. These two market forces, informally speaking, were rather sluggishly matched to each other at the time, and only altered a bit later in the 1960s- which we shall get to in a moment. As the economist might say, the market exhibited equilibrium at a low level. What need there was for change was not because of students, but because of need for more programming time for programmers in scientific and business settings.

McCarthy and Minsky thus brought in their earliest students, including both scholars who completed Ph.D.’s and ‘programming bums’, now called ‘hackers’, who did not care enough about academic degrees to finish them. This neglect of credentials represented a loss for the university Bursar, but not for technological progress. With time to burn and a surfeit of intellectual ability, hackers have been perhaps the most prolific inventors of computing’s process innovations (innovations which

make existing things work better). The occasional exhibition of “hacks”, meaning elaborate pranks for which MIT students are famous, has apparently been part of MIT’s culture for decades. But around 1959, for some undergraduates at least, the practice turned to computing became more regular, as the work was first used for clever exhibitions of intelligence on a computer (159). For some it turned into a way of life, which superseded classes and graduation.

Regardless of effects on academic transcripts, the hacks proved immensely productive for computing per se. Once timesharing was developed, Minsky and McCarthy began to attract hackers enchanted with the ease of access to the machine. However, McCarthy, according to various CBI interviews, believed that he was not adequately respected for his contributions. He left MIT for Stanford in 1961, shortly before the AI contingent at MIT was deluged with research monies. A group of graduate and undergraduate students and professors who studied and used computers coagulated around 1960, two or three years before a larger commercial boom in business computing.

As we mentioned, Jerome Wiesner, President of MIT, helped in getting projects enroute, and gave Minsky and McCarthy student assistantships from JSEP. This was nothing compared to the funding that followed. For several years, their institute was a poor cousin amidst the wealth of the entities within MIT. ARPA had been founded in 1958, as we saw earlier, as a response to the Sputnik crisis. But it was only in 1960 that President Kennedy initiated an effort to improve the quality of command and control capacity in armaments (160). This devolved to the creation of the Information Projects Technology Office, and to vastly improved funding for AI. In 1962, Minsky’s ‘rich uncle’ and longtime colleague J.C.R. Licklider brought about the ‘miracle of Project MAC’, returning bearing gifts, specifically enough money to fund the computational dreams of Minsky’s colleagues and students.

Project Mac, Revisited

The friends and colleagues were renamed the MIT AI Laboratory, which ran for ARPA Project MAC, or Man And Computer. Project MAC was also known, unofficially, as “More Assets for Cambridge” (161).

Project MAC received three million dollars a year, and was independent from the RLE and only nominally part of MIT's Computation Center. The grants were badly needed and well-used, but not solicited: Minsky admits that "I don't recall ever writing a proposal for that" (162).

Project MAC continued to depend on ARPA throughout the 1960s, and to pursue improvements of timesharing and systems, and lasted as such until 1974. It generated, at different times, two major MIT laboratories. Minsky and Papert and their AI cohort seceded from Project MAC around 1970 to found the AI Laboratory. However, they lasted as formal administrators only for two years: Michael Dertouzos joined in 1974 as Director of Project MAC. He promptly changed the name to the more sedate Laboratory for Computer Science, because "MAC sounded like a hamburger" (163).

The Miscellaneous Results of Hacking

Much has been made of the unique culture and personal quirks of the hackers. We will not linger on the topic; Stewart Brand, Steve Levy, and Fred Hapgood have all written excellent accounts. The philosophical base behind hacking was loose and intuitive rather than grand. A fine illustration, perhaps, is that of a famous 'hack', or prank, several years prior to hacking computing, in which a group of MIT undergraduates contrived to lift the complete body of a police car, with a functioning siren, onto the grand dome of the MIT campus (164). It was apparently near impossible, bound to impress, and above all, fun. Thus much hacking was basic and everyday as AI's means and ends are complex.

Unlike the police car escapade, computer hacking has bequeathed the world useful and lasting things. The first principle was that computer facilities should be readily available to one and all, and relatively easy to use. The philosophical axioms which justified hacks included a belief that "information wants to be free", as Ted Nelson said in Computer Lib. In certain cases, it is difficult to believe that the instinct to make computing activity more democratic and freely available was not an echo of the Marxian axiom of putting the means of production into the hands of the forces of production- that is, the workers (165). This idea is bolstered by the fact that the

movement to popularize computing apparently included a number of 'red diaper babies', that is, the children of Old Leftists. Thus, the Leftist challenge to capitalist authority, or any single authority, over machinery and wealth, and the more simple American and Anglo-American tradition of tinkering with machines, were all embodied in hacking.

It is self-evident that the tools produced by hacking- editors, programming languages, operating systems, vision systems and robotics projects- were useful. But in many instances they were meritorious not only in spite of the light-hearted approach to technology but because of it. As a model of computer development, this one was often excellent where certain more hierarchical approaches were problematic. As Richard Stallman- the ancestor to whom all modern-day hackers are avatars- put it in a 1981 guide to the EMACS display editor:

“ The conventional wisdom has it that when a program intended for multiple users is to be written, specifications should be designed in advance. If this is not done, the result will be inferior.

...The development of EMACS followed a path that most authorities would say is a direct route to disaster. It was the continuous deformation of TECO into something which is totally unlike TECO, from the typical user's point of view... I believe that this is no accident... Neither I nor anyone else visualized an extensible editor until I had made one, nor appreciated its value until he had experienced it. EMACS exists because I felt free to make individually useful small improvements on a path whose end was not in sight. ...” (166)

At MIT, hacking on train sets, radios, and remote-controlled boats and cars and the like had been the rage for much of the century. Hacking on computers began almost as soon as it possibly could begin. The aperture to slightly wider access to computing began to open by 1960, and proved quite revolutionary. User-centered innovations such as games, on-line editing, debuggers, and even word processors, seem to have been in large part the result of computing time made available to university students.

McCarthy and Minsky had offered students access to the computing terminals, in the late 1950s before timesharing really existed, and in so doing brought in a few adherents. This provided the first obvious demand for computers with more immediate input and output, which had to be built because they did not yet exist

commercially. In 1961, Lincoln Laboratories donated its used TX-O computer to Minsky and McCarthy and their group, and the RLE was given the earliest Digital Equipment Corporation's first PDP-1 ("Programmed Data Processor). Meanwhile a donation of the castaway TX-O from Lincoln Laboratories, and then the new PDP-1 ("Programmed Data Processor") from DEC, allowed access to be given in earnest to the undergraduates, who were given the grand title of "Systems Programming Group" for the TX-O (167). It was not coincidental that the TX-O and the PDP shared an ecological niche: DEC's founders left Lincoln Labs to create a machine that was basically a commercialization and improvement of the TX-O. Later, a number of the hackers would be hired by DEC itself, as their contributions were adopted into the successive improvements of the PDP machines. Other hackers were brought onto the technical staff of the RLE, BBN, and the MIT AI Lab (which ran Project Mac).

Finally, others would find themselves at Informational International Inc., or 'III', which Ed Fredkin formed as an associated consulting project and small-time hardware developer in the mid-1960s in Tech Square, in the immediate neighborhood of the Institute (168). Fredkin began his career as a SAGE programmer in the Air Force, then moved to civilian life as a Lincoln Labs employee. This proximity led him to the MIT AI Lab and Project MAC prior to moving to BBN and later to establishing his own company (169). Student and colleague Daniel Bobrow noted that amidst the work at these technical centers, Fredkin was busily making plans to move to South America because of fear of the Bomb- which doubtless would be dropped on the militarily significant site of Cambridge, in the event of it being dropped at all (170). After that scare passed, Fredkin remained involved in such AI topics as computer chess, including one program with which he became so obsessed that he carried the related document in his briefcase for three years, until finally the project was taken up (171).

The TX-O, as we recall from the earlier volume of Building the Second Mind, had been developed for the SAGE Project in the 1950s. It looked much more like a radar screen with toggle switches and a console than like what we now think of as a computer. However, gradually the toggle switches would disappear, the keyboard would become more central, and the CRT display screen

would become larger and more important as an I/O device. This evolution was nowhere more pronounced than at MIT. The hackers, who were at the RLE along with Minsky and McCarthy, began writing systems software for it, producing an operating system and TECO, one of the first word processing and editing systems (172). At roughly the same time, McCarthy was developing timesharing, using the computing facilities at Bolt Beranek and Newman in order to be able to fix his own programs. This further widened access for students. Security procedures were non-existent until, eventually, an actual theft took place. Like a number of current hackers, these early hackers would frequently disable, on principle, even innocuous security features designed to keep personal files private. The privacy of individuals' files may have been in dispute, but the common ownership of tools was not. As Levy states, tools were unequivocally public:

“ Tools to make tools, kept in the drawer by the console for easy access to anyone using the machine” (173).

According to Minsky, the operating philosophy at the MIT AI Lab, as the student-professor-machine cohort was soon renamed, was bereft of any sense of 'history in the making' (174). Activities were documented in a fairly haphazard manner, as a researcher can easily see from looking at memoranda that lack even a date of publication, or date of mimeography. Not surprisingly, the bottom-up and hacking research modalities produced numerous and variegated results. It is interesting that no matter how much was documented and even celebrated, other significant creations will remain forever undocumented: “the eyeglasses we made with tiny CRTs projecting solid images before the eyes”, and “the little wireless computer terminal (which was promptly stolen)” (175). Alternatively, other objects were documented but never actually constructed. These include the “Graphical Typewriter”, which Minsky proposed in 1964. The Graphical typewriter would have offered not only the x, or horizontal axis, so to speak, of conventional typing of letters and numbers, but also a Y-axis for plotting (176).

The professed lack of self-consciousness as to history in the making is intriguing because, in fact, so much of this work did make

technical history. The effect is of a chaotic car accident in which the results were swift but creative rather than destructive. Anarchical lightheartedness coexisted with grand ambitions, and grand accomplishments. Minsky tells us that:

“...we tended to assume for better or worse, that everything we did was so likely to be new that there was little need for caution or for reviewing literature for double-checking anything. As luck would have it, that almost always turned out to be true.” (177).

Substantive work emerged from both the ‘hierarchical’ programming of the earliest time-sharing projects and the unmatriculated hackers. This underlines that in computing and AI, one should look at substance rather than style.

While the star of the show is often portrayed as having been Project Mac and the AI laboratory, there were definitely other environments in which AI work was done. These include the RLE, the department of EE, Lincoln Laboratories, BBN, which was closely associated with MIT, the MIT Computation Center, Ed Fredkin’s private concern, Information International Inc., and others which appear more rarely, such as the MIT Electronic Systems Lab, which created the first multiple display system model for CAD (178). We include these as well, and will sometimes be somewhat indiscriminate in referring to the AI and computing devices invented at MIT during this period.

The MIT environment, writ large, contributed enormously to AI in several dimensions during this time period. We may delineate a number of distinct areas, including time sharing, systems and languages; Word Processing and Editing Programs; Chess and Games; and I/O Devices. The Institute also produced robots and vision systems, as well as the more ‘cognitive’ programs, often closely associated with the efforts in embodiment of human-type receptors and effectors.

Time-Sharing Systems, Systems and Languages

Philosophy aside, the idea of AI is only as good as its implementation. This implementation, in turn, was only as good as the time-sharing system which allowed people to carry on their

work, and languages and operating systems which facilitated that usage. Both timesharing and the affiliated operating systems and languages were subject to continuous development throughout the period we are studying.

As we saw, timesharing was practically a necessity for AI. How else to allow endless hours of experimentation and code-writing ? Timesharing works by intermittently doing bits [sic] of each allotted task until every task is accomplished. More technically speaking, “in a multiprogramming computer, several programs may be processed concurrently by switching from one to another in a fixed sequence to permit a certain number of instructions to be performed on each occasion” (Penguin Dictionary of Computers; (179).

As we saw in Chapter Three, McCarthy was not the sole inventor of timesharing. Yet he was, apparently, the first one who proffered the idea of a ‘computer utility’, or “a community utility capable of supplying computer power to each customer where when and in the amount needed. Such a utility would be in some way analogous to an electrical distribution system” (180). MIT’s Long Range Committee on Computer Facilities, shared this opinion, and developed the Compatible Time Sharing System (CTSS), the first of its kind, in 1960 and 1961. Fernando Corbato was the primary engineer (181). CTSS was the parent of the MAC system, which was central to the development of all AI and indeed all computer science at MIT, and which we will now consider.

The MAC system was, in its essence, simply a “supervisory” program in core memory, which put other programs into a highly selective queue. It allowed the partial advancement of the programs inputted, forcing each to procrastinate somewhat and proceed in tiny units. The basis behind the relative alacrity of the Mac system was the fact that the supervisory program had, in the tape drives which constituted its memory, the instructions of each computer language. This meant that the user’s program in any of an alphabet of computer languages- FAP, MAD, MADTRAN (a translator of FORTRAN into MAD), COMIT, LISP, SNOBOL, ALGOL, etc.- was already quite familiar to the timesharing system. The users’ desired programs, whatever their language, were adequately familiar to the timesharing system that the queuing could proceed far more

rapidly. The supervisory program was also a particularly inquisitive and capable executive secretary. As described in one of Robert Fano's papers, it also "handles the communication with all the terminals [that is answered the phones of users calling in, and accepted or rejected their calls], time sharing of the central processor on the part of the active programs, moves these programs in and out of core memory, and performs a variety of bookkeeping functions necessary to protect users' files and maintains detailed accounting of the system usage" (182).

The MAC operating system, and timesharing itself, replaced a human systems operator, who personally and exclusively was entitled to load punch cards into the mainframe computer. He would be replaced by the modern systems manager, who for decades sternly demanded whether the foolhardy users had backed up their data to floppy disks, or turned the computer off without closing all windows and programs.

The Mac system began in 1963 with capacity for ten terminals, and began to accumulate more users, more terminals, greater capacity for remote computing, and a plethora of applications of all varieties. By 1964 when Fano wrote his paper, there were 52 modal teletypes and 56 IBM 1050 Selectric typewriters, located in various places on the campus and in some academics' private homes (183). The MAC system allowed dozens of users to simultaneously dial in to the system and write their texts and run their programs. It thus succeeded the limits of the MIT Computation Center, which by its nature had to be highly exclusive in who it allowed access to programming. This enriched the users- for whom usage was now far less dear and far more convenient. Early implementation of the concept of "dialing in" to a computer, through a 1200 bps telephone connection, indicated that this could work, not only across town but across the world. In addition to the more regular access through the MIT private branch exchange, the MAC system could be accessed very remotely through the Telex or TWX telegraph networks. This sort of access was demonstrated in some European locations, an idea that was revolutionary at the time (184). The MAC system also was a testbed for implementation of innovations in I/O, specifically for the light pen and for early display terminals. We will consider the

I/O innovations, implemented in this project and elsewhere, later in this section of the chapter.

Finally, the MAC system was important because of its role in the larger genealogy of operating systems for research and later commercial computing purposes. The hackers created ITS, the Incompatible Time Sharing system, as a response to CTSS, but did not succeed in getting it established as the primary TS for the Dec PDP 10s (185). During the 1960s, as the MAC system was enlarged and improved, another project which drew upon the experience of Project MAC was introduced. MIT, Bell Labs, and GE were involved in the Multics timesharing project, which was basically derived from the artifact and programming experience of CTSS and the Mac system. Multics itself went in a direction orthogonal to AI. In 1969, Bell Labs withdrew from the project. Multics was subsequently adopted by Honeywell for a secure operating system for computers for the military (186).

UNIX, the brunt of many dumb computer jokes and even more serious criticisms, was developed in response to Multics as an operating system for minicomputers. (The name Unix is itself a response to Multics' name). But perhaps he who puns last puns best. Richard Stallman, the ultimate hacker, developed an expressly free "copylefted" operating system with many UNIX features, which he called Gnu ("Gnu's not UNIX") as a further pun. One might plausibly argue that the entire tribe of operating systems which has powered computers in the late Twentieth century may be traced back to the CTSS and the original Mac operating system.

Languages, Word Processing and Editing Programs

A timeshared environment with many terminals was a fertile petri dish, in which many cultures grew. It allowed the invention and subsequent updating of many versions of LISP, the users of which started adding facilities by which to clarify the flow of data to memory locations in the central memory, set the quantity of memory allocated to the program; and a number of other things that would make programming less a matter of endless tiny details. The programs were also specifically intended to enhance the capacities of the environment. For instance, the LISP programs

developed specifically at the MIT AI Lab (the TS LISP for the PDP-6, for instance (187) were intended to integrate data input from the light pen or the vidisector (the visual input processing device).

Some projects were editors, which facilitated the writing of computer programs. Others- clearly fewer, were text editors, which allowed the typing of prose and its output to printers. These were nothing like the word processing programs of today. Only programmers could possibly begin to understand the cryptic commands. But, then, there were no casual users at this time anyway. The function of the text editors was to input text- that is, to write- but one needed to basically program the machine in various ways in the process. For instance, the user had to specify the printer, had to specify the location in which to store data, and had to enter formatting commands from a list of rudimentary commands. It is hard to imagine that the text exactly flowed easily in such circumstances. The first major display editor (meaning one in which one edited one's text or programs on line) was TECO ("tape editor and corrector"), which was invented with the arrival of the first PDP at the RLE, and then went through consecutive versions for each new PDP. TECO itself had many adjunct programs and programettes [sic], such as the one which justified text or produced a ragged right border. Such things were genuine programming innovations (188).

The text programming and text editors appear to have morphed or evolved into each other like single-celled organisms combining and recombining. TECO thus underwent successive versions, and with some help, finally morphed into EMACS, and a number of other editing and programming tools, a decade later. In 1974, Stallman began to modify TECO, apparently with the general goal of improving it and the specific ends of improving the display processor and clarifying the command set. His changes, in turn, were based in part on existing features of the editor E at SAIL. He did not intend to actually turn this project into an entirely new language, but rather to change some of the features of TECO: "EMACs stood for Editing Macros, before we realized that EMACS is composed of functions written in a programming language rather than macros in the editor TECO" (189). That is, the 'fix' was intended to improve the functions of the macro set. Stallman and

colleagues used this “EMACS” designation for the program before they determined that they were developing a new programming language instead of fixing problems in TECO.

After significant evolution, they ended up with an extensible program editor- or programming language, if you will. The EMACS editor was characterized by a high level of extensibility, which made it attractive for users. Moreover, it provided numerous amenities. It allowed the user to make significant changes, such as redefining self-inserting characters, and to add to and select from a library of functions. It had extensions for reading mail, for editing a file directory, and for reading tree-structured documentation files, and could find files in its index.

The programming style is notable, and perhaps decisively so for the ultimate design of the program. The most modest of people, Stallman has historically not sought material gain for his creations (190). Thus, it is not strange that he allowed the larger hacking community to add to and alter his program. Stallman expressly purported that the emergent and evolutionary development style which he pursued in developing EMACS was key to its success.

EMACS was itself inspired by E, a somewhat comparable editor at the Stanford AI Laboratory; as we said earlier, editors were just one of a number of process innovations which the AI environments in general produced. EMACS was fruitful and multiplied, having been dispatched to at least one hundred sites, according to Stallman, and imitated at least ten times. Several of these adaptations were developed for, or at least later used for, commercial purposes.

ChessPlaying Programsand Games

Games are serious stuff in AI: checkers and monkey and bananas and Tower of Hanoi and, the most lasting game, chess, have all consumed great chunks of time because they are simply such great gedankenexperiments. The AI Lab environment also came forth with games, in the grand sense of the term- that is, in the form of computer chess- and in the form of the legendary Spacewar, one of the more enjoyable but less highfalutin’ versions of computer entertainment. In the sense in which chess is currently played as a

dead-serious manner, it is difficult to even call it a game at all. Regardless of its improbably designation as a game, it was an ideal hack, for a number of reasons. Most of these reasons were the same as those that had been taken up at the drosophila of AI a few years before this- that is, chess was digital, and had a very tiny semantics but an enormous problem space.

Richard Greenblatt, Donald Eastlake, and Stephen Crocker began to create a chess-playing program late in 1966. Their approach was, in their own words, “pragmatic”. Within several months they had developed a working prototype:

“We did not pretend to be writing a general problem solving system, but addressed ourselves directly to the problem of chess...” (191).

This is not the goal of a great deal of chess study, in which the researchers try to achieve results relevant to cognitive science, as observation or emulation. Yet their results were useful and successful. Their program consisted of a search of a problem space expressed as a game tree in which “the branches of the tree correspond to alternative moves and the nodes correspond to positions”. An evaluation ran through the inventory of plausible moves; these were ranked by numerical values for longitudinal merit. The moves were also evaluated in a static sense. MacHack supplemented this analysis with heuristics. Over the course of many plays, fifty or so heuristics accrued, many of them general-purpose, some specialized to certain cases (192). The program played everyone it could find, so to speak, to a total of about two thousand games against players at every level by late 1968. Of these, it won roughly 86% of games facing tournament players (193).

Input-Output and Graphical Devices

The innovations of the third quarter of the Twentieth century are better appreciated if we do not take them for granted. The current desktop and laptop convention, in which the user communicates with the computer by looking at a display monitor and typing at a large movable ‘QWERTY’ keyboard, while intermittently manipulating the cursor with a ‘mouse’, was not a fait accompli or

even the norm in the early 1960s. Far from it: computers were still in the process of evolving from radar systems. They had consoles, toggle switches and buttons, and looked more like an airplane or the control deck of the Starship Enterprise than like the neat small package that the average user knows now. The display of data was a work in progress, too. The best display systems were character-based. TECO, for example, displayed text on a screen, and this text was (painfully and slowly) manipulated by commands from the keyboard (teletype). Thus, even the skeleton of the current system had not calcified by the early 1960s.

Perhaps the most salient difference perhaps is not even in technology as much as in the necessarily lower status devoted to display monitors for computing. As we will see in a moment, display monitors for computing itself did not exist, and had to be imported and renovated for this purpose. One of Marvin Minsky's 1968 papers indicates both a pressing need for decent cheap display monitors, and that much of the display systems in the form of light pens and screens with rudimentary bit mapping, were simply too expensive:

" For special advanced projects, time sharing is incompatible with adequate displays...We recognize that the MAC 7094 system is marginal in many respects and cannot be expected to meet computational needs that would tax the machine non shared. For example, research on animation techniques ... might run into such limitations. Another example; it is technically feasible to implement hand printed input through light pen recognition, but this might overtax the computer unless it had a very special status." (194)

As long as something is not entirely impossible, "too expensive" is usually a matter of priorities. In this case the appellation appears to have referred to both the reluctance to attribute a giant chunk [sic] of space to the buffer for display monitor purposes, and a sheer scarcity of bytes.

The customary IP method of the late 1950s and early 1960s time was to load punched cards, paper tape, or magnetized tapes (vaguely like cassette music tapes without the cassette, but much larger in diameter) onto the computer. One usually received the output by means of paper tape as well, or by printout via teletype, on superwide paper. Display monitors were sometimes used for offline editing of programs, or more rarely for the amazing luxury of

direct input into the computer. Any decent monitor would greatly facilitate this sort of editing. The problem was, there were not any decent monitors. Robert Fano and his colleagues, while establishing the MAC operating and timesharing system, improvised by borrowing a display unit, originally developed for CAD computer-aided design, from the MIT Electronic Systems Lab. The display system, in which input was carried out by teletype at various remote locations, edited online (or offline if the timesharing limit had been reached), was the best man-machine integration which the Mac System offered. However, this early unit did not even share the basic features of the current CRT or the more recent flat screen LCD unit. It was a round porthole of a monitor about a foot in diameter, and it was adapted from an oscilloscope.

An oscilloscope is an instrument used in radar and scientific settings, which measures waveforms and pulses. The instrument's CRT screen shows the course of these phenomena in the form of the amplitude and time between wave peaks. These two data correspond well to the Cartesian X and Y axes respectively. The device used the technology of a cathode ray tube (see earlier) for the visual display. Despite the very different purposes for which it had been originally developed, it was adapted to graphical and character display, and for several years referred to as a 'scope' rather than by the current designation. DEC also developed its own monitors, not far off from the original TX-O 'scope' monitor, and these appeared with the PDP series computers in the AI and other computer laboratories (195).

The displays for the Mac system, which also found their way to Project MAC and the use of the larger AI and CS community, were modified so that they produced characters and could be manipulated with the light pen (196). As we said, the means of visual communication between human and computer was not yet solidified. The uncertainty of the means of HCI was indicated by the surfeit of means by which to talk to the computer. One could use the teletype or the light pen, but also for different programs, one used various buttons and toggle switches and knobs. In 1964, when Fano wrote the report we are discussing, the visual oscilloscope-based display system was connected directly to the central memory of the MAC system (an IBM 7094). To use this highest level of

facilities, rather than just the remote teletype, one had to be in a room next door to the computer installation (197). The early MAC system thus provided more direct I/O than ever before. There was at least the start of a visual display system, with the user interacting with the computer by looking at screen and doing something to manipulate that screen. During the 1960s, I/O began to consolidate into this form, minus some things, such as the mouse and the GUI, that would be considered essential now.

The Lightpen

I/O took a brilliant false start in the form of the light pen, invented by Ivan Sutherland at Lincoln Laboratories in about 1960 (198). The pen is an electronic wand, which could be used to touch the display screen and to draw figures and graphical devices. Technically, the light pen was a fairly simple device. It was a photocell, that is, an electrical device, which produced a binary state that the computer could read, and transmit the bit map of the light which the pen touched on the graphics monitor to the memory. At a fairly low computational cost, the computer tracked the pen's progress. Thus, the figures were relayed to memory- in a manner that was technically relatively simple. In Minsky's words:

“ The light pen facility has, in itself, a completely trivial cost. All is required is a photocell amplifier connected to a flip flop whose state the computer can read- a sense light or a ‘status bit.’ tracking programs based on simple interrupt or sequence break operations need consume no more than a few percentages of the computer’s capacity.” (199)

The Light Pen was used mostly for line drawings, as in the Sketchpad system, rather than for editing. This made sense because it was a more complex technology than the teletype entry of data, and because it was not an ideal “pen” for writing with. The Sketchpad program, which Sutherland introduced shortly after the light pen was developed, was augmented by further civil engineering programs which combined intelligence with graphic functions. The Sketchpad system made drawing figures the labor of the computer, rather than placing that burden on the user. Thus the computer could itself draw lines, rotate figures which had been

inputted, and in the later and more sophisticated applications, calculate stress and physical forces brought to bear in engineering designs. The latter applications include some of the earliest AI work of Larry Roberts, who would later become an IPTO and PARC administrator and champion of the ARPAnet. Sketchpad was followed by subsequent programs which allowed the portrayal of solid, that is three-dimensional, objects- “drawing will be directly in three dimensions from the start. No two-dimensional representation will ever be stored”.

As with other early programs, the user controlled the program by pressing buttons and switches for commands, rather than by interacting with a keyboard. This was of course, before the convention of display monitor and keyboard had been well-established, and visible external buttons, toggles, and switches were a better way to know commands or state than was waiting for the paper tape to punch itself out.

The pen made a terrific splash at MIT. Particularly given how cumbersome it was to communicate through typing in an age of primordial word processing systems, the communication through line drawings that Sketchpad offered was impressive. Used for designation of characters, rather than the engineering and line drawing applications that made up most of its work, it was also perhaps ahead of its time in that it simply required what was at that time the great expense of the bit mapping of the screen. It asked for too much power to be devoted to the buffer space of the monitor. It was sometimes used for text editing or “display sensing” mode, that is, in which “the computer asks whether the pen has seen a certain character or picture element; it has in effect asked the user a yes or no question” (200). If used this way rather than for line drawing, the pen could pose as big a problem as it solved:

“ While this would not be too much trouble where the picture can be divided into a meaningful coarse mesh, or where one wants to draw a curve or graph it would be fairly deadly for applications like Sketchpad, and a problem even for text editing. While a table could eliminate a search, it would have a high cost in memory renovation and in set-up time.” (201)

Minsky is describing a bit-mapped screen, in which each pixel is tied to a memory location in computer memory (in current architectures, in RAM (202)). Such a system is a requirement for modern graphic applications and the Alto-Apple-Windows type PC desktops. While this system was apparently being envisaged as early as the late 1960s, it was at the time simply too expensive because of the look-up table of associations and the cost of the refresh rate.

There is perhaps a further reason why a light pen would be a problem for IP over a longer duration. The light pen, combined with the teletype, are a pair which probably do not work well in tandem, simply because of the positioning of the user's hands. The user cannot sit comfortably and ready to touch both the screen and the keyboard below it. Lifting the hand to touch the screen becomes physically tedious and does not combine well with using the keyboard. One or the other of these IP instruments will necessarily be placed in an uncomfortable way. Such program architectures appear to be distinctly non-ergonomic, as in the case of the HP touchscreen personal computer, which was in vogue, briefly, circa 1985. In contrast, the computer mouse, immediately adjacent to the keyboard, does work because the user's left or right hand is so close to the device which manipulates the cursor on the screen.

But the computer mouse did not yet exist, and would not be widely publicized for a good decade- until the mid-1970s. Instead, the user was given too many things to fiddle with- knobs, toggles, switches all around the computer display screen, if in fact he (almost always a he) had a screen at all. Too many input devices were spoiling the HCI broth. The problem was how to get the keyboard and the monitor as the central IP devices. This consolidation appears to have taken place over the course of the 1960s. The one thing that is really largely unchanged is the keyboard. However much disparaged QWERTY may be, it is persistent. The "scope", or monitor itself, has greatly improved as a piece of hardware but remained basically the same eerie glowing green screen for two decades or so. It is the overall architecture of the monitor as per the larger shape of I/O that has been changed. Minsky's 1968 paper on I/O indicated that the buffer zone allocated to these displays was not that much of the system:

“ The solution usually proposed here is that we abandon the light pen in favor of a non-optical position sensing device. The result is that the computer can interrogate the device, at any time, to find the X Y coordinates of a hand operated ‘cursor’. This raises the same serious programming problem in an even worse form, because one has then to make a test for the proximity of the pointer to the various significant area of the picture: this requires not only a table but a comparison routine. While this would not be too much trouble where the picture can be divided into a meaningful coarse mesh, or where one wants to draw a curve or graph it would be fairly deadly for applications like Sketchpad, and a problem even for text editing. While a table could eliminate a search, it would have a high cost in memory renovation and in set-up time.” (203).

The Improvement of Visual Displays

The problem would remain the same with either Sketchpad or a cursor- invariably presented in quotation marks in the early 1960s. In order for online I/O to be implemented in earnest through a display screen, greater resources needed to be devoted to it. Better visual displays were indeed being developed by the 1960s, and the I/O display monitors were indeed available, as per the vision programs circa 1966. Minsky, and others recognized the unwieldy nature of I/O conventions as an impediment to programming:

- “ 1 There is an important place for a large number of low-quality displays...
- 2 We can tolerate display quality far below military specifications.
- 3 Moving (motion-picture) displays are not essential for every program.” (204)

We have so far considered the more deliberate means of input into a computer, that is, volitional human typing or touching the screen with a light pen. A further possibility is to reconsider just what the input is, after all. There were other ideas being considered as well- for wider inventory of all sorts of IP that were, or that someday might be, being accepted by the machine. First is the idea of making the image presented to the TV camera into the image which goes into memory. By the mid-1960s, the AI Lab had developed a data input device called the TVA television adaptor:

“ Any standard closed circuit television camera can be connected to the PDP-6, without modification, by a single BNC connector. Then a simple program can make a digitized image of selected size and position appear in core memory. Operation is automatically controlled by the PDP-6 priority interrupt system so

that, to the programmer, the core image is automatically read in and maintained.” (205)

Early Handwriting Recognition Programs

The idea of handwriting recognition has been attractive for a long time, but remained difficult to achieve. The MIT AI Lab produced ARGUS, one of the first programs of this kind, and in this case, an application for which the light pen was highly useful. The program learned to recognize handwritten characters, written with the light pen, and would respond to this input by displaying the formal script of the letter in response. Training the program consisted of drawing characters in the raster, and correcting ARGUS’s guesses where necessary. (A raster is a video display which swept beams through a fixed pattern. An image made up of a matrix of points would gradually accrue. Webster’s New World Dictionary).

“ ARGUS is written for the DEC PDP-1 with 4096 words of high speed memory and a parallel drum. It is compatible with the installations at both MIT and BBN. Briefly, the program learns to recognize characters drawn on the face of the cathode ray tube with the light pen. The program may be trained to recognize a particular style of handwriting and a particular character set and the results of this training may be punched off and saved for future use.” (206)

Tracking Eye Movements

One final, and amazingly imaginative, form of input turns the movements of the eye into the input itself. Eye tracking sounds like the fond dream of Madison Avenue, and the bad dream of the paranoid on the street, but actually the idea is not so odd as it sounds initially- on the contrary, Minsky and Papert point out that eye tracking redresses the lopsided nature of interaction between the limited IP channels of the computer and the multifold IP channels of humans:

“ Man can see, hear and touch the computer in many ways and places, but the machine is restricted to receiving information through a narrow bottleneck- usually a Teletype. We have been interested for a long time in redressing this imbalance. This year we were able to achieve an old goal of enabling the machine to look at a person. More precisely: the machine looks at the mans eyes to determine his point of fixation.” (207)

Tracking eye movements was a customary part of cognitive psychology, though, and a program originally developed at Honeywell through a NASA contract was turned into one which displayed text for a reader, and then pronounced, or translated, the word currently fixated upon (208).

Robotics and Vision

The MIT AI cohort initiated work in the areas of robotics and vision- basically, effectors and receptors, very early. Perhaps this was a rational segue from the work typically done at MIT. In contrast, these fields did not begin at Stanford or CMU until several years later and the end of the 1960s respectively.

This precocious research in embodiment appears to have been an unintended consequence of the structural nature of MIT as an institution that focused on engineering rather than the arts and sciences. MIT had been a technology center for decades before AI got started. The institution took on a post-war structure very early, due to its intimate involvement with the Cold War. The university was not solely a liberal arts college- in MIT's case, it never devoted itself to liberal arts at all. Instead it was home to huge numbers of non-faculty- research people, tinkerers, and gadget builders. Moreover, many of these people had started their research questions with Cybernetics, or with radar and the early Cold War technologies of acoustics and electrical engineering. This helps to explain the profusion of artificial receptors and effectors, and people who pursued this research into AI. Thus the structural nature of MIT helped lead to, "bottom-up" AI - creating intelligence from embodiment toward mind. The purportedly bottom-up approach- through tinkering, not with the economic man and decision making on IBM punch-cards, did indeed lead to the top-down questions, in robotics and in vision.

Robotics was initiated at the RLE with the pioneering work of Henry Ernst. MH-1, commonly seen as the world's first truly mechanical hand, was the point of origin for numerous robotics and associated projects (209). Ernst started by obtaining the hardware of his intelligent arm, an industrial robot that had originally been designed for atomic energy work. It was competent, if not at all intelligent.

With a human running its master controls, it could screw in a light bulb or pick up and strike a match. By augmenting its effectors with sensors, its feedback loop was closed and a dumb prosthetic thing began to turn into a smart thing. It was given these “senses” in the form of electrical pads that detected the intensity of pressure against its fingers, and photocells on its fingertips, so that it could know, or rather “see” whether or not the hand was near a dark object (210). Connected electronically to the TX-O, MH-1 advanced as far as the “table-clearing routine”, in which the hand swept back and forth over a table, bumped into things, picked them up, dropped the stray objects into a garbage can, and returned to its sweeping (211).

Further series of programs worked with more advanced robotic arms and introduced far more programmable features. By 1968, the hackers had designed a robot arm which would catch a ball. It worked, perhaps too well: the arm could swing around in one half of a second, and caught people as well as inanimate objects. The arm was eventually fenced off for the protection of all parties (212). The design of a robotic arm took parallel form of the design of prosthetic ‘minds’. The Lab participants examined both the functions of the human hand itself, and the nature of existing notations to describe physical movements. Apparently, existing languages in the orthopedic literature and in choreographic languages were not adequate. The close description of “actions in terms of interactions of position, force, velocity and sensory responses” were not articulated, so the laboratory made up its own (213).

Late in the decade, the AI Lab participants developed a formal language to facilitate the movement of the robotic arm, and the AI Time-Sharing System, as it developed through the decade, was altered to accommodate the robotic limbs of MH-1’s successors. The available robot utility functions progressively enlarged the repertoire of arm movements. The routines included the articulation of a number of arm and hand positions, starting with six expressed on the X, Y, and Z axes; description of the extension and movements of the robotic hand to encompass grasping, rotating, and curling of the hand and of its digits; and the setting of the velocity of the arm movements (214). In this instance at least, one can only do what one can express in a formal language, so the

alteration of languages and the time-sharing operating system itself to accommodate robotic limb movements was necessary.

Notwithstanding the information processing and neurological simulation challenges, the design of sensors from an engineering standpoint was apparently not overwhelmingly difficult:

“...it is quite feasible to engineer a grasping surface with good pressure sensitivity at a great many points by wrapping a coaxial cable connected to a time domain reflectometer. (A TDR is a sort of radar system designed to measure reflected radio waves that are produced in a soft sheathed tube, by any deformations of the wall. The instrument permits 100s of 1000s of points, using a single electrical connection to the hand).” (215)

Then again, this passage does wonders to remind the reader that “difficulty” is always relative.

Some robotics and vision questions might, at first glance, appear to be “exclusively” concerned with robot movement or the acuity of light sensors. But this was not the case with the issue of developing more closely integrated receptors and effectors, with more diverse repertoires of action. This question instead inevitably turned into an issue of “general” intelligence and cogitation.

” There is a larger problem here; how should one represent a machine’s body image ? For the problem of a single not too complicated arm, one can doubtless get by with cleverly coded, sparse, three dimensional arrays, but one would like something more symbolic. And one wonders what happens in the nervous system: we have not seen anything that might be considered a serious theory.” (216).

Minsky and Papert’s 1968 research paper then ponders the simple act of putting an object upon a table, turning around and then turning back to grab the object:

“ We would presume that this complex motor activity is made up, somehow, of a large library of stereotypical programs with some heuristic interpolation scheme that fits the required action to some collection of reasonably similar stored actions. But we have found nowhere any serious proposal about neurological mechanisms for this, and can only hope that some plausible ideas will come out of robotics research itself....” (Ibid.)

Vision

How does one begin with a question as big as vision ? Big Questions often have areas that are diffuse and others that are tight nodes of difficulty. Thus they offer various ways to start- cutting through the node of difficulty, or trying to go at it in the easiest way first, or simply beginning with a vocabulary. Trite and irritating to say, but we may start at the beginning, if only we can find it. In this case the beginning is not only the phenomenon itself, as in humans and other creatures with the capacity of vision, but is also the object of vision as well. In this instance, the beginning is a photocell, a device that can 'see". A photocell is an electronic device, in this case a cathode ray tube, which responds to light, for instance a light pen, by emitting electrical power. This is useful because it provides the fundamental basis for all graphic displays (217). The photocell is only a sensor or receptor, however, and is not per se intelligent. The first efforts in vision gave data on arrays of light, either sensed by a photocell or simply in digital form, and connected these arrays with particular semantics.

Work on vision at MIT started as early as 1962, with Adolfo Guzman and Larry Roberts, with programs that were given photocell data or digitized data, virtual pictures, we could say (218). The data usually consisted of lines, often ellipses or geometrical figures with straight lines, on a blank surface. Such a visual scene was "read" through a camera which presented the data as a two-dimensional array of numbers. The Lab began by using the Vidicon, a commercial television camera, which proved imperfect.

They progressed to a custom camera called an image dissector, which read images projected onto a cathode tube. It then discerned the relative intensity of the light in each point of the image. This camera read light intensity as a floating-point number or as a logarithm of intensity, rather than only a binary value. This produced a much more subtle measurement (219). (Information International, Ed Fredkin's commercial venture on the side of his longtime work at the Lab, developed this tool).

The Lab's 1966 Summer Vision Project dealt with scenes as simple as line drawings of non-occluded figures. It also addressed solids and the backgrounds against which they were portrayed. In such a rudimentary beginning, the goal was to separate out the solid or dark portion of the drawing- "the region", typically- from the background, or "chaos". "Figure-ground" analysis proceeded hand in hand with the description of the schematized and simple shapes, often squares and ellipses, curves, and the like (220).

Like the tree that falls in the forest but still makes a sound, an occluded figure is still a figure. But it may be difficult to see that this is so. To get more computational bang for the buck, so to speak, one needed to wring more knowledge from less data. This led to the stratagem of looking at the Y joins, that is, the meeting-places of the figures' sides. From different sorts of y-joins, one could induce or deduce the nature of the figure partially revealed. This led to a mathematical-computational field called computational geometry, which studied "the mathematical representation of geometrical figures in computing terms" (221).

Within two or three years of the origination of computational geometry, the Lab produced the SEE program, which could address much more complex data. Using computational geometry, which thought its way around hidden faces of figures, it could analyze: "a simple scene, three stacked cubes, with partial occlusion; outer edges; optical detection of edges, including the more subtle aspects of edges, including slope changes, simple steps, and highlight or crack". Minsky and Papert observe that some real world knowledge was indeed required to analyze such scenes, but SEE and other programs could indeed progress quite far without solving more general issues in vision as a part of "overall" intelligence (222).

At the same time, other programs were developed which considered surfaces rather than only lines. A 1966 program called 'regions1' took a predicate as a function which defined the nature of a region, or shape. That is, the input was:

"1 an array filled with numbers, which are intensities read from the vidisector
2 a point (inside the array)
3 a predicate, which defines a region.

The output was:

Output: a list of two things;

1 number of points on the boundary

2 unsorted list of boundary points. (A point is a list of two numbers, namely their bi-dimensional coordinates;

3 also, marks in the array the points belonging to the region found (with a - sign);

Purpose: given a starting point, finds a region around it satisfying a given predicate. Marks this region, and returns its boundary." (223)

The program could be given a predicate- that is a shape written as a function, and then depict the 2 dimensional shape that would satisfy that function. The Summer Vision program also began to do what people do, that is, to save time and energy by knowing what different specific objects look like- knowing how to see the object, so to speak:

" A collection of pictures is beginning to be formed on microtape; this tape contains several objects which were seen by the vidisector and written with VDTAPE. They consist of a big collection of integers which represent the light intensity of a given point.

Sussman tape #3 currently contains:

WHITE CUBE a cube;

CYLIND WOOD a wooden cylinder

CLOTH COUCH a piece of Seymour's couch." (224).

SEE, the later project, could discern the full shapes of partially occluded bodies, although its capacities were restricted to scenes formed by straight lines, without shadows or noise are present (225). Other work on vision addressed ambient environment and the eye's perception as much as the abstract reality of a cube or a collection of geometrical objects:

" Even if the surface is not of uniform color and texture, there is still much more that can be done with a monocular picture. Lawrence J. Krakauer is developing a system that analyses scenes such as a bowl of fruit. The project begins by locating and analyzing illumination maxima; it appears that from their intensity, shape, [and] behavior one can in many cases, distinguish full from shiny surfaces. Local maxima connected by an illuminated band are likely to be on the same object, and Krakauer is testing some other heuristics for associating highlights with edges." (226).

Despite this progress, it was recognized that vision is so highly developed and so intimately interwoven with real-world knowledge, that it is not simply a matter of “scaling up” projects which use Y-joints to induce the geometrical shapes of rectangles. To “see” things that people saw, rather than just the idealized scenes that the computer was presented with, could be very different. Many concessions were made in the form of transformations of real pictures:

“...The picture is subjected to a sequence of transformations; each transformation is intended, in turn, to produce a successively more abstract representation until, finally, one obtains the desired description of the scene. Typically, such a sequence might be:

- 1 remove noise (by clipping, smoothing, etc.);
- 2 enhance features (by boosting gradients, etc.);
- 3 extract features (finding edges, vertices, etc.);
- 4 group features into objects (by regions, parallelisms, etc.);
- 5 identify objects (by partial matches, etc.).” (227)

Instead of simplifying the fact of vision, more knowledge appears to have made it more complex. To really understand a scene in a human sense is highly context-dependent and involves much pre-processing of the data:

“ One must cope, for example, with

- 1 direct line of sight occlusion of parts of objects
- 2 shadow occlusions that depend on the directions of lighting
- 3 highlights
- 4 reflections
- 5 textures
- 6 decorations
- 7 many other interactions between visual features and spatial forms.” (228)

The intense, perhaps deceptive, knowledge-relatedness of vision helps to explain why one thing the researchers did not do initially was to construct an artificial retina. Their project appears to have been a-vision, which involved a-mind, rather than a re-creation of the human apparatus. Reverse engineering rather than AI as cognitive science: that is, this form of vision concerned the geometrical calculations needed for defining figures, not the internal properties of the eye. The vision is in the equations and the head, rather than about the physiology of sight. At the end of the

1960s, neurology and the physiological edge of cognitive science came into the picture with the residence at the Lab of neurologist David Marr (1945-1980).

Marr, who died tragically young, is generally credited with the discovery of the “primal sketch”, meaning the purported initial view of a given object in the form of a crude geometrical sketch, rather than the refined picture which we perceive as “seeing” immediately. But otherwise, Minsky and Papert offered specific reasons not to build artificial retinas:

“... the operations of the vertebrate retina and the subsequent image processing is not nearly so well understood as is generally believed, and we do not think the time is quite ripe for taking such a step in hardware...” (229).

Edge detection was a clever device for synthetic vision for simple scenes. But for more complex scenes, further computational deduction was needed to limit the variety of objects which any given combination of edges could add up to. Constraint propagation, invented by David Waltz as his MicroWorlds dissertation project, was able to analyze more complex scenes. By analyzing the format of the junction of edges of figures, the possible geometric figures which could be depicted were progressively winnowed down. The number of sides and the visual appearance of corners in different geometric three-dimensional figures differ greatly, and from the various shapes of these junctions, the possible array of figures represented is progressively reduced. That is, the constraints upon an initially large number of possibilities are gradually propagated. Waltz’ constraint propagation seems to be a precise visual equivalent of limiting search, or to put it more emphatically, rather the visual instantiation of search. Constraint propagation is a pruning device quite appropriately opposite the search forms of the time, although again, the cognitive emulation elements in this technique are not salient. (230).

Thus, the environment was fruitful: a cornucopia of nifty things tumbled out of the Laboratory. Artifacts that are documented include editors, robotics projects, vision and neurological research, eye tracking devices, haptics sensing projects, and machine learning and Classical AI, that is, cogitational projects. Given the

multiplicity of bottom-up approaches to intelligence, the later emergence of agency theory from MIT is no surprise.

Chapter 6. Carnegie-Mellon: From GPS to the Physical Symbol System and Production Systems

As we saw in the introduction to the 'Belle Epoque' of AI in the 1960s at the three major research institutions, the "high road" versus "low road" distinction offered by Edward Feigenbaum during the mid-1960s provides meaningful demarcations. Feigenbaum discerns these two distinct paths regarding research on generality. The first sought a universal route which would be ostensibly applicable to as many circumstances as possible.

The best example of approaches on the high road was, naturally, Newell and Simon's General Problem Solver:

" Researchers on the low road eschewed a general problem solving system in favor of exploring specific complex domains and tasks in order to (1) test whether ideas methods developed so far in AI could be used in significant problems and (2) identify new issues for AI basic work." (231).

This search for the 'high road' to universal heuristics for intelligence dominated the early 1960s:

" By 1965, the search for a general system for developing machine intelligence had become one of the most pressing issues in AI. The area of problem-solving taken in its entirety since the middle 1950s constituted the AI search for generality... Researchers sought a formal system so general that all problems however represented internally, could be translated into the form of proving a theorem in this system yet so specific that general proof methods could be developed for it..." (ibid.)

In spite of a decade of effort in this area, Feigenbaum could comment in 1968 that:

‘we lack a good understanding yet of this problem of generality and representation’.” (232).

Beginning a couple of years into the 1960s, there was a reorientation of the field toward semantics, that is the representation of knowledge rather than syntax in specific areas. Such distinct paths as inquiry into vision or movement as such (low-road), versus general problem-solving techniques is not as deterministic as it might appear to be. It does not mean that the fruitful research concerning a particular mode of cognition will have no further implications. For instance, the low-road philosophy at MIT engendered more general knowledge representation formalisms in the case of frames, in the early 1970s. A great many actual applications prompted by the research of Feigenbaum himself appear to have derived from the low-road inquiry into precise cognitive activities. To employ a familiar axiom, the road not taken can be reached later. (That is, in the case of Dendral which we will consider later in this chapter, experimental evidence in problem-solving).

The Origins of AI at the Carnegie Institute of Technology

As we have seen throughout this work, ARPA was the eight-hundred-pound check-writing gorilla of the first three decades of AI. But considerable research took place before and beside its patronage. Infrastructure for the study of Complex Information Processing was in place at the Carnegie Institute of Technology in Pittsburgh for several years before anyone uttered the words ‘artificial intelligence’. Herbert Simon and colleagues put this structure in place at the Graduate School of Industrial Administration, or GSIA.

Simon was recruited in 1949 from the University of Illinois at Chicago, where he had been a political science professor who specialized in organizational behavior. Fluent at an early age at both the profession and the scholarship of university life, he had already written Administrative Behavior, various articles on municipal management and other topics, and was about to publish Public Administration (coauthored with Donald W. Smithburg, Victor Alexander Thompson). The GSIA was formed with a six-million dollar

grant from the Mellon Foundation, and Simon was brought on as Professor of Administration (233). Simon himself credits Newell with the skill of 'grantsmanship', but for the first six years at GSIA, before Newell joined the CIT, Simon and his colleagues carried off the magic act. He showed something of a Midas touch in attracting research money: he was given a grant of forty thousand dollars by the Carnegie Foundation (234) and various grants by the Ford Foundation and the Office of Naval Research (235). As with the other universities, IBM provided the 650 in the mid-1950s. Simon duly recorded this construction of institutional infrastructure in his autobiography, but made clear that he dislikes the petty aspects of politics. He devoted an entire chapter to high-level university politics, with a worldly disclaimer entitled "Why I am not a University President".

Grants were won on the basis of the brilliant research, but a brilliant research administrator was also essential. CIT was blessed with such an administrator, Al Perlis, who devoted himself to having a computationally and fiscally comfortable environment for research. CIT's Computer Science Department was only formally organized in 1965, through Perlis and Simon's efforts. Prior to that, the AI courses had been taught through Simon's professorship at the Business School. Thus Simon's early students, Edward Feigenbaum and Alan Newell among them, received their doctoral degrees in Business Administration. But during the early 1960s ARPA declared the GSIA Computing Center its first "center of excellence" (236). By this time, the institution was fully supportive of what was already by that time a longstanding and prolific endeavor. Neither MIT nor Stanford received grant monies on the unrestricted terms which CIT won with this title. The "center of excellence" designation proved highly fortuitous for the fusing of the cognitive psychology and AI. At all points in the research program, Simon's sheer enthusiasm won converts. Edward Feigenbaum, who would have a similarly evangelical effect on others in AI years later, recalls:

" In the mid-1950s, I was an undergraduate engineering student at Carnegie Institute of Technology... I signed up for a seminar Simon was giving- Mathematical Models in the Social Sciences....
... the final session after the break brought an unforgettable moment. Herb Simon came into the class and said, 'Over Christmas

AI Newell and I invented a thinking machine.' Puzzled looks from students contemplating an oxymoron. Machine ? Thinking ? Thinking Machine ? What could that mean ? He laid on the table some copies of an instruction manual for the IBM 701 Computer. I remember staying up all night, absorbing that manual, undergoing a conversion experience. What followed in class was LT [the Logic Theorist] and IPL 1 [Information Processing Language 1] ... and then it was obvious what to do for graduate school (work with Simon and Newell) and what to do for the summer (learn to program real machines at IBM)." (237)

Of course, Edward Feigenbaum was himself amenable to enthusiastic responses to grand intellectual projects. The passage points to a number of traits which were fundamental to the construction of the research program in AI, computer science, and psychology: willingness to get into details, even when programming was mundane, slow, and very difficult; belief in the scientific process, and a joint concern with psychology and computing.

The Physical Symbol System Research Program

" In his Origin of Species Charles Darwin discusses the advantages and disadvantages of island environments for engendering new species. There is little doubt in my mind that the support and enthusiasm of the Carnegie campus community for the tender green shoots of the new information processing psychology played a major role in their survival and growth until they were sturdy and vigorous enough to compete with other species of psychological theory that flourished on this continent and in Europe." H.A. Simon, Models of Thought p.xv.

As portrayed throughout this work, early knowledge engineering in AI- the Classical version, we could say- takes the historical proposal of Empiricism in philosophy seriously, as a construction project. AI proposes that the mind's processes are rational and meant to take in the environment in an unbiased way. This is not exactly the case, as in the well-known instances of the primal sketch in visual information processing, or biases in memory due to rehearsal effects. However, particularly in the case of cognitive processes- thinking in the pure and colloquial sense of the term, it is a decent working hypothesis that this author does not consider obscured by time. As interpreted in AI, this has traditionally meant that the mind may legitimately be studied in relative isolation from the other faculties (e.g., perception, and receptor and effector capacities). For

modeling purposes, cogitation could thus be easily depicted on a digital computer, with computational input and output rather than the noisy 'real thing' brought to us by the senses. Empiricism has furthermore contended that thinking may be depicted by the movements of tokens. The nature of philosophy is argument with words rather than argument with empirical demonstration, but modern instrumentation has allowed the Classical mind's convictions to be transformed into science.

Newell, Shaw, and Simon (NSS) began their work in the 1950s with the idea of computation as a model for expressing and testing psychology. This theorem carried forward Simon's scholarship into other fields as well. His articles on economics and organization theory developed the ideas that Man typically satisficed rather than optimized, meaning that they searched within a limited ambit for the best solution to their needs. Likewise, people are typically bounded rather than omniscient in his knowledge. These theorems inevitably led toward psychology, and this turned into AI. The working premise for NSS' research in psychology and AI is known as the Physical Symbol System hypothesis. Newell and Simon's acceptance speech for the 1972 Turing Award contains an excellent concise presentation of the PSS:

" A physical symbol system consists of a set of entities called symbols which are physical patterns that can occur as components of another type of entity called an expression (or symbol structure). Thus a symbol structure is composed of a number of instances or tokens of symbols related in some physical way, such as one token being next to another. At any instant of time the system will contain a collection of these symbol structures. Besides these structures, the system also contains a collection of processes that operate on expressions to produce other expressions: processes of creation, modification, reproduction, and destruction. A physical symbol system is a machine that produces through time an evolving collection of symbol structures. Such a system exists in a world of objects wider than just these symbolic expressions themselves." (238).

The PSS proposes that such symbol systems are viable methods for both explaining and replicating intelligence. In this way it is what is often referred to as an essentialist model, in which essential basic components are proposed. The term is usually meant as a slur, although such systems may be quite intricate. Where the essential doctrine of cells as the basic unit works in biology, and that of

atoms works in physics, and the germ theory works to explain disease, the essential doctrine of creating viable intelligence through symbol structures in software, running on computers, explains the replication of intelligent activity. Their method of demonstrating the way in which input was processed and returned as output- to use the most gross reference possible- was the digital computer for the machine, and thinking-aloud protocols for the people (239). The terms in which the PSS was subject to demonstration changed, though. In the early 1960s, Newell adopted a definition of learning as demonstrated through performance of a task rather than through altered adaptation to given stimuli, which was harder to measure.

The PSS is a model, with the advantages and the limitations of models. It is a hypothesis, which means that it is falsifiable and can be tested. It is small and clear enough to test, but it is a subset of the world rather than the world itself. Some features are stylized and indicated only schematically, and this is as well as can be done at any one instance. Particularly before 1970, the generalizations which pertained in PSS work included portraying sensory experience as digitized input and output. This proved inescapable, although Newell and Simon's recruitment of Raj Reddy from Stanford seems to indicate that they wished to broaden their scientific project's foci.

Newell and Simon and their colleagues at Carnegie-Mellon developed a grand body of scholarship based on Classical thinking. They turned these words into models, and the models into machinery. They conducted experiments in cognitive psychology; the results of these experiments were called "protocols". From these results they derived models of mental processes, which were tested by running models of psychological functioning on computers. The human results and the computational models were used iteratively to draw up an increasingly refined model of various aspects of mental functioning.

Academic work can take place at any of a number of different gradation points between simply science, in which only causal mechanisms and processes are sought out, and simply engineering, in which only replication of desired results is sought. The work of Newell and Simon and their cadre of colleagues covered the full

breadth of this spectrum, in a number of disciplines. They were thoughtful, like philosophers, but also productive, like engineers. They thought through carefully ideas such as the need for parallelism (virtual or in hardware) when considering AI embodying physiological processes. Yet they were prolific and productive in constructing good working knowledge representations.

Economics' Blind Turn

Ideally, the examination of the nature of human symbol processing could have extended itself to other sciences which directly addressed human decision-making. Simon had been an economist before he began work in AI or psychology. Indeed, the implicit psychological questions of microeconomics had inexorably led him to psychology and hence to AI.

Regrettably, the numerous discoveries of AI did not lead Simon and his colleagues back to economics until the 1990s. The colossal foolishness of the economics profession's emergent decision to ignore Simon in the early 1950s, is a prime example of a non-optimal decision- even a non-satisficing one. The disapproval of the economics community compelled Simon to become an economist-in-exile. He did not return to the field for over a decade. And when he did, he may have been the one who laughed last, for he received the Nobel Prize in Economics. Economics' loss was several other fields' gain. As in any search process, certain nodes had to be pruned. The choice of this particular pruned node is certainly as fine an indication as any of that search is not necessarily optimal, just as there is no free lunch.

This approach simply idealizes input and output in the form of computational data- at first punch cards, and later digital input in the form of computer tape reels. Human beings use wetware rather than this sort of 'dry' input and output- we have sensory receptors and effectors in the form of the senses and our capacities for speech and writing and other physical actions. But even if digital IP and OP are simple and schematic, research has to start somewhere and use models somehow. The NSS (or NS) and large PSS approach does schematize some of its data, but it does not seem that this compromises its internal integrity.

The Scientists as Cognitive Psychologists and Human Problem Solving

No one faults the violinist who cannot play the flute as well. Musicians typically are known for their performance in one instrument. This is true in academia as well. The scholar who manages to work in parallel in two or more disciplines is extremely rare. Newell and Simon accomplished this feat, in different ways. Immediately relevant to us among the many fields they worked in are AI and computer science, and to a lesser extent cognitive psychology. Simon continued his controversial scholarship in economics, as well as studying the philosophy of science, the psychology of scientific discovery, design, and organization theory. Without making scholarship subordinate to big science, they obtained steady research funding and the requisite departmental recognition, and colleagues in the form of both students and professors. They built a first-rate psychology department, and a school of computer science. CMU also grew a bumper crop of resident professors and a harvest of distinguished alumni. The ambit of the PSS model in examining mental phenomena seems to have been, and continue to be, nearly all-encompassing, reaching thus into many academic fields. It reminds the author of Hinduism's hundreds of Gods- one for every possible human trait or occasion. As AI's 'Hindus', so to speak, Newell and Simon and their intellectual "children" appear to have integrated every possible problem-solving and knowledge-representation heuristic into one giant intellectual edifice.

Even as we struggle against circularity, what we believe influences what we try to prove. In the early and middle decades of the Twentieth Century, psychology typically tended to reduce mental questions to physiological ones, and in the process nearly negate the weight of cognition. But just after the middle of the century, Cognitive Psychology experienced a Renaissance. Newell and Simon were very much a part of this. They had to be a part of it, or even central to it: Artificial Intelligence built on a Behaviorist metaphysics would have been absurd. Ignoring the psychological processes which they were keen on reiterating would have been like ignoring one's twin. AI seems to have always functioned for them as a dual-edged scientific and technical tool. NSS relied on Russell and

Whitehead's logical theorems and on DeGroot's study of chess players for the very earliest work in 'Classical AI'. They soon were joined in their insistence on studying problem-solving. This is not the right place for a digression into the Cognitive Revolution, but the development of a vigorous field of study in cognitive psychology parallel to AI was essential to the latter's success (240).

The AI research produced at CMU is better understood when we observe the close participation of Newell and Simon in the Cognitive Revolution in psychology. The AI world thinks of them as computer scientists, and that is their major significance for this book. But both participated in the professional life of academic psychology even in the 1950s. Newell joined the major United States psychologists' association in 1952 (241). Simon's academic home base was located in CMU's Psychology Department starting in the 1970s- and he garnered as many academic 'second homes' as a billionaire has houses. Their involvement with the precepts stated at the announced beginning of the Cognitive Revolution in the mid-1950s was more than simply close. It was their movement as well. Simon apparently became irate at George Miller, E. Galanter and Karl Pribram, for writing a seminal book, Plans and the Structure of Behavior (1960) which Simon claimed contained a good many of his own ideas (242).

The subject matter of cognitive psychology includes such mental phenomena as imagery, short- and long-term memory, problem-solving, search, and learning. This incorporates historical questions of the nature of thought, but contemporary cognitive psychology intensified its consideration of such questions. Especially with the help of computer modeling, asking old questions could yield new results. The Introspectionist school at the turn of the Twentieth century had the will but not the means: they lacked a rigorous analytical method (243). What you can look at is only as fine-grained as the tools for observation will allow. Now such tools were available. This is another instance in which certain sciences cannot emerge at all until adequate scientific instruments are available.

The intuitions of folk psychology may provide useful starting points for more rigorous research; but this is still only a starting point. In order to rise above homilies, the cognitive scientists in mid-century

adopted a policy of minute and exacting study of their object of interest. They were, like the Behaviorists, fairly obsessed with performance, and insisted on performance and (in the case of the AI-cognitive psychologists) computational feasibility before accepting internal state. The Behaviorists' methods is conceded to have been one of their best traits; it was their chief 'selling point' at the start. One could even say that their methods were too good for their own ideas.

But the method wasn't too good for both the Cyberneticists and the Cognitive Psychologists (244). Among the latter, Newell and Simon and their colleagues adopted an exacting iterative method which equaled the Behaviorists' insistence on objective evidence. They gathered protocols of problem-solving processes for semantically austere fields such as checkers and cryptarithmic. Then they postulated a problem-solving heuristic and closely reiterating it with a computer program. The raw data of the protocols themselves, complemented by programs, were taken with the gravity and authenticity of the Rosetta Stone- which they may have been.

The AI programs robustly demonstrated the internal state associated with human problem solving, and other psychological phenomena. They did this using demonstrable output associated with equally demonstrable inputs and internal processes. Their results, however, were obtained using the methods of their ardent adversaries. Newell and Simon stated that this was the way to reconcile the vicissitudes of behaviorists and Gestaltists, and that they were "natural descendants of both" (245). It is not surprising that this magnanimous sentiment was not returned. The bravery of the scientist is putatively located in his or her willingness to accept the risks of falsification. But this does not mean that anyone wishes to see the precepts that they really wish to see vindicated proved false !

By showing that one could use rigid 'Behavioristic' criteria to show that internal state and other taboo objects, the Cognitive Revolutionaries simply twisted the knife. Newell and Simon were not distracted by lack of acceptance from this group. It has been observed that B.F. Skinner never responded in writing to Noam Chomsky's scathing review of Skinner's ideas. It is certainly equally

significant that Skinner never publicly argued with Newell and Simon, whose voluminous works refuted his. Skinner's three-volume autobiography never mentions either of the two.

The psychological research conducted by the research program of Newell and Simon et al., broadly speaking, was as much concerned with psychology as with AI. The research addressed highly focused concerns within the discipline's research issues. Some of these, such as the serial position effect (the study of the relative efficacy of memorization of a list of nonsense syllables), are windows to characterization of the nature of information processing itself. Simon and Feigenbaum, for instance, used this problem as a grounds for demonstrating the human usage of information-processing heuristics such as breaking a long list into two more manageable chunks (246). This is a canonical issue in cognitive psychology.

The program in AI thus turned into a full-fledged department of psychology, with a strong emphasis on the processing of visual information. Later on, at the very close of the 1960s, Newell and Simon recruited Raj Reddy to CMU from Stanford, where he continued his ongoing work on perception (vision and speech understanding) (247). Performance is a good way to prove intention: this would widen the scope of inquiry at the university and would eventually lead to one of the world's most distinguished program in robotics. Since this work's emphases are on knowledge representation rather than the full and immense world of robotics and embodiment, we will leave it at this point.

Cryptarithmic, Chess, and Other Drosophila

" ...Trying to learn to use words; and every attempt
Is a wholly new start..."
T.S. Eliot, Four Quartets.

One of the ironic features of the early years of any scientific enterprise is that insights that will later come to seem patently obvious and trivial are among the hardest-won. This is evidenced in the repeated and insistent recourse to the same problem over a long period of time. In the case of discovery of basic concepts in

computational problem-solving, learning, and knowledge representation, the substance upon which the insights were won, or the battle joined so to speak, were several austere experimental problems. Sciences need their respective experimental bases, and these problems- chess, checkers, cryptarithmic, the Tower of Hanoi, and the logical proofs of the original Logic Theorist- were the drosophila for early AI and for cognitive psychology (Drosophila is the scientific name for the common fruit fly, a species which is used extensively in biology experiments). The metaphor was coined by Herbert Simon or by John McCarthy, or independently by each (MOML).

Cryptarithms are “arithmetic problems in which letters have been substituted for numbers and which is solved by finding all possible pairings of digits with letters that produce a numerically correct answer” (Webster’s Ninth New Collegiate Dictionary). They are to algebra as pig Latin is to Latin: engaging, if not rigorous or intellectually respectable. But like pig Latin, cryptarithms are not meant to be either. Problems like these- chess, checkers, cryptarithmic, the Tower of Hanoi, and the logical proofs of the original Logic Theorist- were the proving ground for early AI and for cognitive psychology.

These problems share certain features: perhaps most notable is their compactness and essential featurelessness. Explained in no more than a few minutes, they are as sparse as a monk’s cell or a library thesis-writing carrel. The spareness and absence of noise provides a certain translucency. These ultimately were transparently simple systems, which helped to focus attention on the entwined scientific goal of understanding problem solving and search, and the technological goal of reiterating these.

The minute and exhaustive attention to protocol records of human beings working through these problems continued for more than a decade. Newell published articles exploring problem-solving heuristics through cryptarithmic as late as 1968 (248).

The Progeny of Pandemonium

Pandemonium, the program which Oliver Selfridge had designed at the end of the 1950s, was doubtless influential. It consisted of dozens of shrieking demons, as they were called, which responded to particular procedure calls in what might be called chaotic, or, alternatively, emergent. It is definitely the basis for the present-day idea of demons, small sub-programs which undertake narrow programming functions such as automatic messages that the email recipient is away for a week (249). Beyond that, this program seems to be the primordial form of agents and of the blackboard problem-solving technique.

Allen Newell, as we have already seen, attributed great importance to Selfridge, and Pandemonium was part of this legacy. Selfridge, in turn, was influenced by the Cyberneticists and their biological approach to intelligence. As we saw in Building the Second Mind: 1956 and the Origins of Artificial Intelligence Computing, Selfridge had checked the page proofs for Norbert Wiener's Cybernetics. Selfridge had been a roommate of Walter Pitts before this and had clearly been concerned with a biological model of intelligent activity, in which information processing is distributed through numerous receptors. In a 1962 conference paper, Newell reflected upon the design of Pandemonium. In so doing he provided the first analysis advocating the blackboard:

“ Metaphorically, we can think of a set of workers all looking at the same blackboard: each is able to read everything that is on it and to judge when he has something worthwhile to add to it. This conception is just that of Selfridge’s Pandemonium (Selfridge 1959): a set of demons, each independently looking at the total situation and shrieking in proportion to what they see that fits their natures.” (250).

Newell considered the blackboard a remedy to the proclivity of programmers to visualize and hence design programs in an unnecessarily sequential form. Despite the apparent natural tendency, logical clarity did not require a sequential timeline or sequence of steps. To avoid both redundancy in information accessible to subroutines and an overly sequential structure, the blackboard is proposed, or specifically the isolation of concurrent routines combined with access to a common data structure. This differs from the speech understanding topography in that the “common data structure” is replaced in current depictions of

blackboard architectures with a common solution space, and heterarchical segmented knowledge (data) structures.

Meanwhile, Newell's 1962 analysis of Pandemonium influenced the eventual creation of production systems and of the later OPS system (251). In both of the latter, the control mechanism at the primitive level is the condition-action rules or if-then statement-granted, a less provocative image than the "shrieking demons" (252). Several years later, Simon explicitly articulated the blackboard model, again in a theoretical essay (253). Discussing human problem-solving protocols, he refers to the habitual generation of sub-goals, the incremental interim solutions to which are held in a 'blackboard' in 'permanent or relatively long-term memory', during the course of addressing the problem's subgoals. This usage resembles the one currently presented as a blackboard architecture.

The demonstrable influence of Pandemonium on Classical AI suggests that the rediscovery of parallelism in representation in AI in the late 1980s was a 'retro' movement that was heralded as prescient. It was actually an old horizon, neglected and then rediscovered. But peering into the evidence of the past can sometimes be just as good as premonition.

By the 1960s, the General Problem Solver incorporated a simple form of learning through **discrimination nets** (254). This simply meant that the sequential runs of a particular search are stored so that the efficacy of various paths is recorded; when the program has failed in a particular node, it records this (255). The final work on GPS was that of George Ernst in the late 1960s, on slightly more robust domains, such as trigonometry (256). The General Problem Solver project was formally concluded in 1967, and was succeeded by production systems.

The magnitude of the opus of the General Problem Solver in NSS' work goes beyond this as well. The essay at discerning basic, generic mechanisms for solving problems was Allen Newell's life's work. In Newell's last interview, he even refers to the SOAR program of the 1980s and 1990s as "GPS done right" (257). Carried on in their specific incarnation by Newell and Simon and later many

other people at Carnegie-Mellon, these evolved into production systems. Carried on in its more general sense to far different ends by hundreds of other people in dozens of places, the avatars of the General Problem Solver were very much in evidence in later AI.

Creativity, Scientific Discovery, and Control Processes

The scientific and technological questions which cannot be answered at a given time are perhaps as important in defining an era as those which can be and are answered. The means for computational reiteration of more than one human information processing capacity was such an unanswerable question for a good twenty years into AI's lifetime. Independent forays were made into robotics, vision, and tactile capacities, but these were not combined with cognitive information-processing. Lacking the component parts for embodied AI, as it is referred to, AI practitioners could theorize as to the nature of information- processing in the body. But they could do this only theoretically. In practice, especially at CMU, the design of problem-solving computing programs habitually truncated the nature of IP and OP into the human totality. In the real world, there is a multiplicity of such inputs and outputs. In the AI world for the first decades, there was only a schematic representation of IP and OP. In Logic Theorist and GPS; world is entirely controlled, because there is simply no other way to do it.

This did not prevent anyone in AI, and Newell and Simon in particular, from approaching topics theoretically. In actuality, the world does come flooding in at us, in many modalities at once. We learn from acting in the world, as Heidegger said. Yet one does not have to buy the entire phenomenological package- that learning is so embedded in particular circumstances that knowledge representation is not possible. Indeed, well before there was any possibility that a computer program could 'perceive' anything sensory and 'think' about it as well, Newell and Simon were writing about this very topic.

Other theoretical considerations of issues which were at the time outside the frontier of practicability include the issue of creativity. Simon addressed this more than once during the 1960s. In 1962 he, Newell, and Shaw published a paper asserting that creativity was

indeed a fit topic for Artificial Intelligence (258). The topic had as yet not been subject to much in the way of scholarship, being instead the refuge of a cult of mysticism. This may be why the paper contains fewer scholarly references than usual. At the time, a defense against conceding that computers could do anything interesting was to say that they could not possibly be artistic or creative. A bit of contemplation will indicate that this topic lends itself well to the metaphor of search or problem-solving. Moreover, judged by performance at least, the Logic Theorist was highly creative. At the end of his initial creativity paper, Simon admits that this really only pushes the mystery back toward other, tougher questions, such as how to characterize the initial state and the goal state. The initiative of debunking the myth that creativity cannot be understood at all, and that computers cannot emulate it either, was important regardless. The initiative was akin to the more general one of indicating how AI could indeed emulate not only the creative processes, but also affective states. Simon was not only compelled to address these issues unilaterally, so to speak, but practically was also willing to be provoked to respond when told that his goals were not possible. In 1963, cognitive psychologist Ulric Neisser made the tactical error of inciting Simon by asserting the impossibility in principle of computers holding multiple goals or having any kind of emotion. This was met with a paper on just this topic, several years later (259):

“ Similarly, the work that Barenfield and I did on chess perception published in 1969, was aimed at refuting the claims of Tichimirov and Poznyanskaya 1965 that a computer scanning a chess board was incapable of grasping chess relations as Gestalts, as these masters could. And a major motivation for my continuing interest in models of scientific discovery has been to show that contrary to the claims of phenomenologists, programs could be designed to discover laws and invent new concepts. Constructive proofs like these were far more effective in answering criticism than rhetoric, however eloquent, would have been. (” Simon 1991, p272). Simon, H.A. “ Scientific Discovery and the Psychology of Problem Solving”. Models of Discovery. Boston, MA: D. Reidel. 1977. (Originally published 1966).

Creativity later became a topic of much concern in both AI and cognitive psychology (260).

Scientific Discovery

If creative problem solving with drosophila could be modeled, so could something of a more closely defined nature. Simon and Newell became concerned with the idea of scientific discovery at about the same time that they first thought of AI as a testbed for emulating creativity. Their first paper on this theme, in the MOT anthology, was originally published in 1962. This was followed by more consideration of the topic, and later by a good deal of scholarship on scientific discovery in particular on the part of Newell and Simon's students during the 1970s and beyond.

The product of Western rationalism could, as we might expect it to, conduct the ultimate rationalist task: that of scientific induction. (Deduction, as we will recall, means the pulling of new knowledge out of existing laws, scientific and otherwise. It is important but perhaps less of a scientific high frontier than induction, in which generalizations are brought forth out of existing data). Scientific discovery was of concern to Newell and Simon as early as the start of the 1960s. This brought both historical scholarship concerning discoveries, and computational modeling of scientific discovery as one of the forms of problem-solving, to Carnegie-Mellon. The first insight, as Simon mentioned in his early papers on the topic, is that discovery is a form of problem solving.

Prior to the re-starting of cognitive science circa the mid-1950s, scientific discovery had been considered a "Eureka !" moment. Close consideration of the topic was muddied by reverence toward the mysterious matter of creativity. But once one really looked at it rather than just saying that it was the eternal mystery, the issue became much more perceptible, and more interesting, rather than a clinical piece of meat. The exact circumstances of the crime, so to speak, included numerous questions for historians of science. But the questions were also central to computer people seeking to really push forward the possibilities of AI. Just as interesting as the scientific issues are those which involve the rendition of the same protocols of human cogitation into computational form.

Herbert Simon and his colleagues Pat Langley, Gary Bradshaw, and Jan M. Zytkow, developed programs which can carry out various forms of scientific discovery themselves. The first major result on

this line was Bacon, still the most famous program of this type. It was named after Francis Bacon, the sixteenth-century scientist and first notable exponent of scientific induction (261). Like any number of things that initially appear impossibly complex, this one is based on a simple idea. Bacon conducts scientific discovery by starting with a nodal nugget of heuristics, and then attempting to apply them repeatedly. Scientific laws are embodied as numerical relations between values- or to put it another way, the law, essentially, is a constant value between variables. (Later versions of the program may take on nominal values, that is name attributes. Most of these are as values, but values also can be formatted as predicates). Discovery, or Baconian induction, is data-driven: that is, it represents its data in terms of data clusters. It represents observations that have occurred together, and has rules for noting regularities between values which characterize a given phenomenon. The repeated verification of a given relation between values is the central means for verifying a law. The system is implemented in a production system language PRISM, that is, one which uses a condition-action syntax.

Bacon, like EPAM or GPS or various MYCIN or Dendral-type programs, lived on for a long time- we could say that it inspired a virtual pork barrel of further programs. Bacon, like EPAM or LISP or GPS or various programs such as MYCIN or DENDRAL, lived on- rather lives on- a nearly human lifespan. Some of these were scientific, as in programs which emulated scientific verification or were frames for understanding scientific paradigms, as in the work of Paul Thagard and colleagues at Princeton and later at the University of Waterloo in Canada (260). The formal rendition of scientific revolutions allowed Thagard and company to come up with a much closer statement of changes in scientific models than had been presented in Kuhn's still-revolutionary work, The Structure of Scientific Revolutions. The topic of scientific discovery was invigorated by the work at Carnegie-Mellon, and took off like a galloping horse (263).

Bacon also begot sons and daughters, named after itself, at CMU. As one of the early successful production systems in a fielded application, Bacon bolstered the production of more, many of them scientific and technical. Production systems in fields less exalted

than scientific discovery were introduced and used in numerous applications such as verification, diagnosis, i.e., probabilistic diagnosis of diseases as in the system Oncocin.

Production Systems

In the late 1960s, the GPS program was retired and the next model for information processing introduced. Newell and Simon were aware that GPS' control structure did not take after the psychological evidence of their protocols, and developed a different model, which more closely resembled protocols (264). Their next project established the viability of computational representation of a problem-solving design for a more robust and life-like domain. Production systems, or 'productions' are one of AI's canonical forms of knowledge representation. Production systems are composed of the primitive structure of condition-action rules, or IF-THEN rules. The terms are interchangeable; the IF segment of the rule is also called the 'left-hand side', and the THEN segment the 'right-hand side'. This molecular level of granularity is a series of conditions which may 'fire' upon designated stimulation, or earlier conditions being met. Above this molecular level, production systems are hierarchical, and therefore more complex:

" A production system consists of three parts: a) a **rule base** composed of a set of production rules, b) a special buffer-like data structure, which we shall call the **context**, and c) an **interpreter**, which controls the system's activity....

" The **context**, which is sometimes called the **data** or **short-term memory buffer**, is the focus of attention of the production rules. The left-hand side of each production in the rule base represents a condition that must be present in the **context data structure** before the production can fire."

" Finally, there is the **interpreter**, which like the interpreters in all computer systems, is a program whose job is to decide what to do next. In a production system, the interpreter has the special task of deciding which production to fire next." (265).

Were the context, or short-term memory buffer absent, the lack of a control structure would impose an inevitable tradeoff between volume of knowledge in the rule base and efficiency in responsiveness in the system itself. Except for toy domains, search must be constrained by an imposed control hierarchy; otherwise, there would be combinatorial explosion.

Production systems were not a new idea, but an entrepreneurial interpretation of an existing one (266). Problem-solving structures using condition-action rules are too generic for that. The concept, but not the term, was suggested as a syntactic innovation in mathematics by Emil Post in the 1940s (267). This is usually cited as the first reference. Even the scholar, Herbert Simon demurred (268), pointing out several antecedents, discovering the concepts of rules which fire upon designated stimuli in psychologist Otto Selz' theory of problem solving (1924). The simple sequence of a condition-action rule bears a structural similarity to the stimulus-response connection in Behaviorist psychology. The similarity is straightforward: "The stimuli are the conditions that trigger the response". This similarity remains even if simply drawing this analogy takes both production systems and Behaviorist psychology's tools far out of their native contexts. Closer to home, it is readily evident that the resemblance to Chomsky's transformational grammar (1957), also a series of rewrite rules, is more than superficial. Post's introduction of the construction to formal logic was brought to string-processing languages by the early 1960s, and later to various GSIA dissertations around 1965 (269). If the idea had been a useful instrument in engineering, it had not been appropriated for cognitive psychology at the time. Simon credits computer scientist Robert Floyd, also at CMU, with first introducing the concept to him (270).

The virtues of production systems are self-evident. The knowledge architecture is highly parsimonious, in obedience to Occam's Razor. That Newell and Simon were engaged in the search for the cognitive equivalent of the atom is an orientation emphatically clear in their collaborative and individual work. Because of the generality of its features, production systems may be incarnate as strong, domain-specific conditions (left-hand side), or alternatively in highly hierarchical form. Either declarative or procedural knowledge may be embodied in this medium. This is perhaps the beneficial flexibility of a weak method. Although conceived at a primitive level, the production systems approach is highly scalable in both the direction of many conditional clauses required for one result to take place, and in the 'vertical' dimension of many rules to fire (with fewer conditional requirements). While intelligible at a primitive

level, a productions approach is scalable and amenable to varying levels of hierarchy.

Modularity is probably intrinsic to an atomistic architecture. Because both left- and right-hand sides are modifiable, both individual clauses and the causal structure of the entire program are incrementally changeable. Thus modularity confers amenability to learning. Insofar as intelligence may be seen defined as adaptability to an environment, this amenability to alteration would seem to be a generic desiderata for 'intelligent' systems. Alternatively, this feature could be construed as a disadvantage insofar as verisimilitude is concerned. This depends, of course, on what constitutes verisimilitude. Should one assert the unmeltable holistic rather than atomistic nature of representation, modularity might be seen as unrealistic. To pursue such a critique, one could simply borrow and apply all of the extant critiques of AI on the ostensible basis of atomism, beginning with a dubious look at Hobbes' declaration that all cogitation is but ratiocination. Engineering efficiency could be construed as a cognitive science detriment, depending of course on interpretation. The usage of a production rule structure lent itself well to the embodiment of substantive realms of declarative knowledge. In addition to their obvious merits for research, production systems were the technical basis for the commercial problem-solving software referred to as expert systems.

The DENDRAL Project and the Breakthrough to Domain-Specific Methods

Dendral, the first major application of a production system to a complex scientific domain, would change AI and open the way toward expert systems. According to Bruce Buchanan:

" It [the Dendral Project] opened up a whole new world... All the stuff that we read about in the philosophy of science we were able to begin to operationalize, as we say now, and it provided a means for understanding in this context what we had to mean by simplicity, if we were going to test the simplicity of hypothesis. We had to be very precise; we couldn't just wave our hands ... Here was an opportunity to create something de novo and see the product in action. And I think everybody around the AI Lab was feeling the same way." (271).

In the early to mid-1960s, the orientation of AI was syntactic. As studied in this book's prequel, search involved toy problems instead of substantive bodies of knowledge. The limits of computer memory remained severe, making search over large spaces computationally expensive. The inventory of different sorts of blind syntactic search included means-ends, the A* algorithm, best first, and depth or breadth-first.

All of these addressed search in general without pinpointing specific instances beyond the pertinent problems. That is, usually heuristics were studied without domains:

“ There was a good deal of work on theorem proving and representing knowledge in an axiomatic system. Many of those people felt that they were on the track of general purpose mechanisms. (272).

Tacitly, such approaches endorsed an accepted common wisdom of intelligence as a generic capacity, independent of embodiment in real life in real activities and varied circumstances, and indeed, much AI consisted of theorem proving and general-purpose mechanisms.

The DENDRAL experiments, initiated in 1965 by Joshua Lederberg, Bruce Buchanan and E.A. Feigenbaum at Stanford, began to change this orientation. They opened the way to AI's contributions to real domains, and to programs that could be implemented for such projects. The idea of a general learning mechanism persisted: Allen Newell's scholarship decades later endorsed the idea of a universal but low-level general mechanism. But the insistence that that mechanism be embedded in pragmatic fields proved to be a blessing to AI research and to cognitive science.

Strategic initiatives are as valuable in intellectual life as in politics. This is particularly so if the rhetoric is reinforced with substance. This axiom is epitomized by the Dendral Project, a series of research projects in the computational emulation of scientific reasoning. DENDRAL used information on rules of molecular structures to derive new formulas (273). These experiments, the earliest results of which appeared in 1968, were notable in their practicality and dependence on a domain expert from outside

computer science. The experiments resulted in the practicable Heuristic Dendral and Meta-DENDRAL systems, as well as fielded applications in the late 1970s and to commercially viable products in the decade after that.

Moreover, Dendral made an impact on AI's image. Regardless of the merits of its practitioners, AI suffered at the time from a misbegotten reputation as a trumped-up and occult endeavor. The 'Big Project' initiative of Dendral was a successful strategy for showing that AI could succeed at a 'real' domain (274). It succeeded in producing fine first approximations of search in dense, complex scientific domains. Moreover, it was decisive in pulling the field in the direction of understanding actual 'expert' problem-solving in such domains.

The Birth of "BuchananandFeigenbaum"

By 1965 Edward Feigenbaum and Bruce Buchanan, known during the decades of their collaboration as "BuchananandFeigenbaum"(275), had arrived at Stanford, from Berkeley and Carnegie-Mellon respectively. Feigenbaum, by his own admission, was no longer interested in the EPAM project (276). His new work with Buchanan and Lederberg quickly drowned out any fleeting ennui. His meeting with Bruce Buchanan proved to form a lasting work team.

Bruce Buchanan became an AI researcher rather than a philosopher almost inadvertently. As a philosophy graduate student at Michigan State University in 1964, he tried to get a summer job at the Rand spinoff System Development Corporation. Someone at SDC thought that his intellectual character was more suited to academia than to industry, and sent his application to Rand, where it reached Ed Feigenbaum. (At the time, Feigenbaum was resident at Rand during the summers). Buchanan had mentioned that his dissertation topic was scientific discovery, a topic that enticed Feigenbaum. Buchanan was swayed by Feigenbaum's systematic and careful thought processes:

“...I was intrigued by someone who was clearly thinking so much about what I was saying... on the basis of essentially one long phone conversation with him I packed up and spent the summer at Rand” (277).

The two worked developed “a program that would learn from examples as a prototype for theory formation”. Feigenbaum invited Buchanan to join the project that Feigenbaum and fellow Stanford professor Joshua Lederberg were starting on scientific discovery and theory formation. Buchanan “decided to give it a year” and moved to the Bay Area. His Stanford appointment stretched into two and a half decades (278).

The roots of the Dendral project are highly original- and yes, it is rocket science. Lederberg, a Nobel laureate in chemistry, sought to produce evidence of life on Mars. He meant to do this by means of analyzing soil samples from Mars, where the presence of organic compounds could indicate that life did exist or had existed. He anticipated that this information could be obtained remotely with a real-time computer on board the next Mars mission. He obtained a NASA contract to plan means by which to procure such evidence. The contract funded the Instrumentation Research Laboratory at Stanford Medical School, at which he continued his work (279).

Unfortunately, the initial plan to use geologic data from Mars proved unworkable. The physical constraints of the time precluded sending a computer with real-time signals to Mars. (Apparently, on the next Mars mission, the proper measuring instrument, a mass spectrometer, was indeed on board). Ultimately, Dendral did not analyze the signals retrieved (280).

Feasible or not, the project proceeded. It wasn't really about Mars any longer, because the information needed would not be made available anyway. But it did become an AI project. Anticipating vast volumes of data, Lederberg wished to contrive a computational means of analyzing the chemical composition of soil substances- non-Martian this time. The relevant technology was mass spectrometry. This is a branch of physics which “deals with the theory and interpretation of interactions between matter and radiation” (Webster's Ninth New World Dictionary). Spectrometry refers to the analysis of the images emitted by an instrument- the

spectroscope- when it measures or records the structure of different compounds. A spectroscope is “an instrument that produces spectra which result from the use or production of or relate to electromagnetic radiation or closely associated phenomena”, *ibid.*

Lederberg had already thought fruitfully and systematically about the generation of new spectra. In 1963, he had invented an algorithm for the systematic generation of numerous plausible chemical syntheses (281). This facilitated the generation of new compounds, but did not help with their analysis. This it merely delayed, rather than eliminating the bottleneck, which was now located in the tedious art of analyzing the plausibility of each of these compounds using mass spectrometry. The topic of crystallography, like spectrometry, is sufficiently possessed of regular, systematic and tedious features of search for matching patterns that it had been suggested as a computer programming task as early as 1953 (282). Dendral was slightly different, and we are looking at it for its role in scientific problem-solving, but actually the general topic of analysis of such data is meat and potatoes for AI.

ARPA shared NASA's interest in funding the work, and so Lederberg, Buchanan and Feigenbaum took on the collaborative effort of finding algorithmic or heuristic means of emulating mass spectrometry analysis. They named this task the DENDRAL Project. The Dendral Project consisted of an onslaught on the fine art of analyzing the results of spectrographs performed on these innumerable syntheses. Buchanan and Feigenbaum and the others who joined them (283), first wished to elucidate the search processes of Lederberg and his colleague Carl Djerassi. (Djerassi is of course also famous for inventing the birth control pill (284)). Then they needed to represent these processes in a program which could classify the criteria according to which compounds were judged plausible or implausible.

The setting for the work also idiosyncratic. The AI participants had their own office in addition to Lederberg's lab, in a residential house, owned by Stanford, on the northern fringes of the Campus. The programmers set up shop in rooms originally meant to be upstairs bedrooms, and were quite comfortable. The Dendral Project

was changed to the Heuristic Programming Project, or HPP, in 1972 “to reflect a broader scope of research.” After the researchers found more formal offices, the house itself was moved in its entirety to a new spot on campus and turned into the Women’s Center (285).

As a bold project of both applied philosophy and as software engineering, the work was brutally difficult, if exciting. Moreover, at least initially, the project was not welcomed with great enthusiasm from the AI community itself. Instead, the team faced skepticism in the beginning. AI’s practitioners were accustomed to doubts as to whether what they did was really science, which had been defined in the field as addressing general principles of thinking, not the specific nature of thinking in one scientific field. They asked Buchanan and Feigenbaum whether what they were doing was really AI:

“ [By the middle 1960s]...” The community being [sic] well-defined. The research problems, I think, were not so well-defined. In the large, anything that had to do with mechanizing intelligent behavior was fair game. And there were a number of people in this community working on a game plan. A number of people working on logical inference and mechanizing theorem proving, and sometimes people were doing game playing through theorem proving...
...we felt a good deal of difficulty in explaining the relevance of chemistry to AI... during a talk I gave at MI-5 [a conference on Machine Intelligence]... somebody began criticizing the amount of chemistry that I felt I needed to introduce. And I will forever be grateful to John McCarthy for saying, ‘Just listen, will you?’” (286).

The project’s early organization in the mid-1960s, Feigenbaum recounts, brought with them consternation and confusion. This was succeeded, eventually, by the reorientation of the project toward the domain knowledge rather than the procedural specifics of general search techniques- an insight he describes as an ‘epiphany’ (287). Feigenbaum’s insight was facilitated by a lecture Allen Newell gave while visiting Stanford in 1968. Nevertheless, fleshing out insights is not an easy thing. The knowledge elicitation part of the project- that is, talking to chemists and engineers- was just as difficult as the knowledge engineering, that is building the program. Buchanan indicates that the project was very much about human beings as well as solving the problem without thinking about how people would have done it:

“...Feigenbaum came out of CMU, so he had a good deal of knowledge about the work on human information processing. And the Lederberg-Feigenbaum hypothesis partly underlying DENDRAL was that what a trained mass spectrometry analyst knew was what the program ought to know. So there was that transfer from human knowledge - thinking - into machine terms. ...what came to be known as knowledge engineering depended very much on understanding what it is that a trained chemist was thinking about, not necessarily the individual inference steps, but what patterns was he looking for in the data, and how did he link those patterns with partial conclusions?” (288)

The Dendral project proceeded slowly while the team interviewed the scientists who did this analysis:

“ Pressing expert chemists to formulate rules about mass spectrometry... proved to be an arduous process”, and indeed one which persisted for years as the chemists themselves were compelled to try to figure out what those rules were. But excruciating was not impossible” (289).

The elicitation of this knowledge in the course of the Dendral work sought out both the declarative semantic material concerning the mass spectrographed fragments, as well as the procedural clues which would provide computational economy and a control structure for the architecture (290). From afar, the domain looked more impenetrable than it actually turned out to be. It proved possible to derive two sets of condition-action type rules for the analysis. One set constrained the imminent combinatorial explosion which Lederberg’s algorithm implied, and the other tested automatically the still-unwieldy number of compounds which resulted from the somewhat tamed algorithm.

“ I believe the perception was that those other groups were dealing with general principles of thinking or reasoning or representation, while we were dealing with mass spectrometry. We knew that we were trying to represent the principles of mass spectrometry in such a way that we could extend the scope of the program. But there was a lot that was very specific to chemistry and what we were doing. The generator was not a general purpose, hypothesis generator. It was a generator of descriptions and chemical structures, and it wasn't good for anything else, except in the wildest stretch of the imagination.” (291).

Heuristic Dendral, the resulting program, first appeared in the Project laboratories in 1968 (292). Heuristic Dendral could

implement economy in both the generation and the testing of new chemical syntheses. The three sequential segments of Dendral included a **planning** stage in which constraints inferred from the interviews with the chemists were imposed to preclude prohibitive computational expense; a **generation** stage in which Lederberg's algorithm was embedded; and a **testing** stage which computationally simulated the spectrometry and provided rules for assessing the fragmentation results. The rule bases for the planning and testing stages are distinct bodies of knowledge, although both were expressed in the form of production rules. The series of condition-action rules were apparently originally composed without any particular order. This made the control structure problematic when the rules reached a critical mass. The entire system was greatly clarified for Feigenbaum by the more disciplined and hierarchical control structure of production systems, which Newell and Simon were working on at the time.

When its performance was assessed relative to that of the humans on whom it was modeled, Heuristic Dendral did quite well. According to the Knowledge System Lab's own report:

" Its performance rivaled that of human experts for certain classes of organic compounds and resulted in a number of papers that were published in the chemical literature. Although no longer a topic of academic research, the most recent version of the interactive structure generator, GENOA, has been licensed by Stanford University for commercial use." (293).

Heuristic Dendral was soon applied to the new chemical domains of insect hormones, antibiotics, impurities in manufactured chemicals, organic acids in human bodily fluids, and products of marine animal sources, among others (294). The new domains, structurally similar, used the same search architecture. Later new application domains included internal medicine.

Yet the Dendral project had a wider import than this. The movement toward domain-centrality, or knowledge rather than search, took place in other centers of AI as well. Knowledge representation, rather than search in toy domains, became the focal issue in the three major centers of AI research during the next several years. The cognitive scientists at Bolt Beranek and Newman produced

schemata; Minsky suggested frames, Roger Schank at Stanford introduced scripts, and Newell and Simon published their thousand-page magnum opus, Human Problem Solving (1972), including close treatment of production systems. Dendral itself was the first AI project to illustrate what became EAF's often-invoked axiom, "In the knowledge, there is the power" (295).

Dendral altered the landscape by initiating a breakthrough to domain-specificity. Knowledge beyond simple toy problems was difficult to study given hardware limitations in early 1960s. But also it was also difficult because of the simple fixation on different search techniques. By proposing that 'in the knowledge, there is the power,' Buchanan and Feigenbaum led the shift from emphasis on general mechanisms toward knowledge. This project also provided evidence that symbolic representation could be used to emulate much more robust sorts of cogitation, while debunking the insistence on generic algorithms for intelligence per se. The close affiliation between specific realms of medical and later engineering expertise turned AI into "empirical inquiry".

The cadre of 'knowledge engineers' who joined the Dendral project and later projects were enthusiastic and even missionary. In addition to infectious zeal, they provided appropriate substance; their demonstrations worked with a real Big Problem. They succeeded by taking what Feigenbaum called the 'low road' in cognitive emulation's questions, circumscribing the particular goal, closely confining the domain, and using varieties of formal, well-articulated knowledge that lent itself to codification. (The "high road", on the other hand, sought universal and more basic forms of understanding of commonsense phenomena; see definition earlier in this chapter). Moreover, in a research-grant environment in which a working demonstration is worth a million dollars, they produced working demonstrations of highly pragmatic scientific tasks.

Chapter 7. Stanford University and SRI

The Origins of AI at Stanford: The Prehistory of Silicon Valley

Stanford University has become synonymous with Silicon Valley. But from the perspective of AI, the field's Silicon Valley origins were at both Stanford itself as well as at its nearby and closely associated think tank, SRI. For a long historical period, both professional engineers and hobbyists have been thick on the ground in the Bay Area. AI's original leaders, with the exception of San Franciscan Allen Newell, came from the other side of the United States. (And Oliver Selfridge, 'AI's Fifth Beatle' as we called him in the first book was originally British). However, the environment at Stanford, and later at U.C. Berkeley ('Cal'), has been more than encouraging. Credit for the origins of the Silicon Valley phenomenon should begin with the Stanford professor, Dean of Stanford's Engineering School between 1944 and 1958, and later the Stanford Provost, electrical engineer Frederick Terman, "the Father of the Silicon Valley" (296).

More general predilections toward technical innovation in this region, in the Valley and beyond, were established much earlier (297). The prospective wealth of the region has attracted skilled workers, extractive industries, agriculture, and investment in real estate, railroads, every sort of infrastructure, and financial services since the California Gold Rush began in 1848 (298). Terman and other academic entrepreneurs have been brilliant and opportunistic exponents of these circumstances, rather than starting out of whole cloth.

By the turn of the Twentieth century, San Francisco boasted more than its fair share of electrical engineers, high-school hobbyists, and bankers interested in promoting their inventions. Others sought to turn skills in hydraulic engineering into gainful employment, following the demise of mining using waterpower (299). California is a large state, and still more engineers were kept busy establishing electrical power across its vast terrain. This inevitably meant steady demand for electrical engineers and for components. Companies to fulfill such projects were found in San Francisco, Oakland, and Palo Alto by the start of the First World War. The demand for electricity for both California and the West was so steady that Frederick Terman and his students were kept busy with such projects thirty years later.

San Francisco and its environs were the site of many 'firsts' in the earliest wireless telegraphy (Morse code) transmissions, in the re-deployment and improvement of that technology as radio signals, and in the usage of radio signals for broadcasting. This list includes the first ship-to-shore wireless [telegraphy, or Morse code] transmission received in the United States, in 1899 (300); the first wireless station on the West Coast, installed by the U.S. Navy at nearby Mare Island in 1904 (301); the first 'organized' radio station with scheduled programs, currently known as KCBS (302); the largest amateur radio club in the country (303); and the development of the amplification and speaker systems which allow us to listen to radio by loudspeaker rather than earphone (304).

The Port of San Francisco itself was the site of many experiments in shore-to-shore telegraphic signal transmission, which was subject to ongoing improvement by Guglielmo Marconi and his wireless telegraphy company. Working in Palo Alto, engineer and inventor Lee De Forrest came up with the three-element vacuum tube, or triode, an essential precursor to modern electronics (305).

Technology does not proceed by hard work alone: someone has to foot the bill. The great wealth of California has often settled in residences in San Francisco and the Peninsula- putting venture funding in proximity to R&D. This makes the extraordinary concentration of tech innovation more than coincidental. In some of these cases, Palo Alto was the location of choice because of the presence of Stanford engineering graduates. But the sheer profusion of inventions, and their location all over the area, should indicate that innovation itself is a more historically indigenous phenomenon .

This propensity to innovate, to borrow a phrase from the economists, was salient without a break throughout the century. Stanford's administrators invested in electronic companies as early as 1909 (306). The trend continued consistently with the encouragement of new enterprises in electronics by Frederick Terman, from the very start of his teaching career in 1926. Terman encouraged his engineering student David Packard in the formation of Hewlett-Packard, maker of acoustical electrical devices, in the proverbial garage in Palo Alto in 1938 (307). These are the most

famous, but there were others as well. The Varian brothers, Stanford professors Russell and Sigurd, developed electronics equipment, most notably the klystron and more notoriously the radar detector (308). They were likewise close associates of Terman but were also, like Hewlett and Packard, local boys stewed in the local hobbyist culture. Before he went into academia, Russell Varian was one of the developers of the first electronic camera tube, built at Farnsworth Electronics in San Francisco in 1927 (309).

Thus much of the Bay Area's innovation was indigenous to the region itself rather than only to the universities. The celebrated Stanford-Silicon Valley connection in particular was not as dynamic prior to the mid-1950s. The custom of university-business relations at that time was indeed largely devised by Terman, and carried through then- and now- by a clearly stated ideology. He regularly sought the placement of students in local technology companies, and the pursuit of research which would be both financially and intellectually rewarding, sponsored by those companies. He emphasized the highly selective hiring of entrepreneurial professors, carrying out research initiatives which he called "steeples of excellence," in narrow and strategic scientific and engineering fields. As Terman put it:

" Steeples that are narrow and high shape the skyline better than structures that are broad and low." (310)

These steeples included fields such as microwave engineering, solid-state electronics, and later computer science, all of which were both politically and financially appealing.

This approach altered the typical profile of the elite preWar liberal arts college, with its emphasis on Attic Greek plays and the Great Works. But it certainly paid the bills. Moreover, the Terman model presaged the development of the modern Postwar university, in which those bills were paid in part by grants from defense and other government research agencies, and by contracts with private corporations (311). This model was complementary to, but not synonymous with, the 'multiversity' as U.C. Berkeley chancellor Clark Kerr called Cal's continuous adoption of new schools and scientific research projects.

We saw in the last chapter that the novelty of MIT's singular devotion to engineering was central to its progress in the Postwar research world. Stanford had actually begun as a liberal arts college, but the latter was swamped by the testosterone of engineering disciplines and other institutional innovations in the 1940s and 1950s. The Stanford Industrial Park (more commonly referred to as the Stanford Research Park) was established in 1953. It was an antidote to a restrictive university endowment that forbade the University from ever selling its many thousands of acres of land. A legal loophole allowed the university to lease out the land and build on it instead, and in the process the rolling golden hills turned into tracts of modernist buildings and concrete. They paved paradise- and once they started, they seemed to find it very difficult to stop. Commercial concerns, with parking lots and maintained shrubbery abounding, arose on such land, located around three of four of the university's perimeters. With the exception of the increasingly luxurious Stanford Shopping Center, these were mostly concerned with with technology. The residents of the Stanford Research Park have included Varian, Hewlett-Packard, Ampex, and Lockheed's new Space and Missile Division. Later clients have included Xerox Parc, Interval, Andersen Consulting, doctors' offices and law firms.

Meanwhile, in the mid-1950s, activity in every technical specialty was related to the new Postwar technologies. These arrived in the greater Bay Area, and especially in the area of Santa Clara a dozen or so miles south of Stanford. National computer companies, including IBM, GE, Sylvania, and General Precision set up shop in the area during the decade, employing 45,000 people (312).

While some participants in this electronics brigade arrived at the Industrial Park, William Shockley, the most famous and infamous figure among them, declined to be closely associated with Stanford despite Terman's offer (313). Shockley located his firm in Mountain View, a dozen or so miles southeast of Palo Alto. In reaction to his apparently imperious personality, his engineers revolted in short order, splitting off like frantic paramecium. From these splinterings [sic], especially the so-called Traitorous Eight, numerous

semiconductor firms including Intel, were born, as we saw in Chapter Two (314).

Meanwhile, Stanford itself benefitted from both the corporate environment and from government R&D contracts, just as the former businesses also benefitted from Cold War R&D. The military-industrial sciences were appropriately photogenic, attractive to corporate donors, and fast-moving to be accorded high priority in the postwar scientific world. The growth was striking. In 1946, Stanford's total government contracts, defense and all, totaled only \$127,000. By 1956, DOD obligations were \$4.5 million, and by 1966 about \$13 million excluding the Stanford Linear Accelerator (315).

Finally, by 1967,

"...Stanford had climbed to third on the defense contracting list, and the top of the national rankings in electrical engineering, aeronautics, materials sciences, physics and other hot fields" (316).

The unmistakably business-minded environment might have provided an engineering university culture which could not wait thirty seconds for ideas to be worked out for the requisite R&D. On the other hand, the university's economic circumstances were such as to increasingly permit, at least theoretically, freedom from worry about outside funding. The actual outcome of the university's location in Silicon Valley was a mixed bag according to the individuals who were doing research. Edward Feigenbaum and Ted Shortliffe, to name two figures, fit very well into the academic-corporate mix, and responded deftly to the opportunities it provided. At the opposite end of the spectrum, John McCarthy engaged in the minimal exertion needed to receive and keep funds from the IPTO and ARPA, while also performing singular, world-class research.

2. Establishing Computing At Stanford

Brewed together with a traditional liberal arts psychology department and an increasingly pre-eminent medical school, computer science was thus born into privileged surroundings. The presence of a great computer science department at Stanford

University may seem so natural now that one may assume that it was born with the university itself. Such was not the case. The formation of this academic department was an uphill battle, waged over a number of years. The appearance of the department was due to design rather than Providence. In 1964, Stanford's Department of Computer Science was nominally inaugurated as a part of the much older Electrical Engineering department. Computer Science became a separate department only in 1985 (317).

The roots are actually much deeper. Frederick Terman had begun to contemplate the formation of a computer science department in the mid-1950s. He did not call it this, because the term itself did not exist. Terman was provoked in part by IBM's proposal to give universities a 650 computer for free, in exchange for teaching scientific and business computing. In 1955, Terman and Stanford Provost Al Bowker hired Louis Fein, a computer consultant at the Stanford Research Institute, to investigate contemporary computing and report on its viability as an academic field. Fein spent one year studying the academic field as it then existed at just a handful of universities, including Dartmouth, MIT, CMU, and Wayne State University in Detroit. Fein's report concluded that "computer science, not yet the computer itself, was a discipline worthy of study by the university". In his CBI interview, Fein asserted that this was the first usage of the phrase "computer science" (318).

Fein's report, and Terman's other inquiries with engineers, led to a 1957 decision to establish an academic study of computing, specifically in the form of a computer service bureau like that which was being established at the time at MIT. The appraisal of computing as an object worthy of academic study was not confirmed by consensus- certainly some saw it as a service department rather than an engineering discipline. The same year, Douglas Engelbart offered his services to Stanford, saying that he could teach computer design courses. Perhaps he wrote to the wrong person: a Dean responded with a short letter explaining that:

"...Since Stanford was a small school and was striving for the highest quality academic disciplines, and since computers were definitely a service activity, that there was no planned possibility for them to bring in computers into the engineering curriculum" (319).

In 1957, Terman hired George Forsythe, Stanford's counterpart to Carnegie Tech's administrative mastermind Al Perlis. Forsythe almost singlehandedly started the department. Originally a mathematician, he had become a meteorologist during the Second World War, and had later learned about computers while employed by the National Bureau of Standards (320). Forsythe's skills at building and leading an academic institution were unsurpassed. He recruited many people to join the Computer Science section of the Department of Electrical Engineering, often with excellent foresight into future achievements. Computer science gained Terman's favor, and joined this roster. Computer Science formally commenced as a separate Department, inside the Engineering School, in January 1965 (321).

Despite Stanford's gold-plated sheen and well-maintained campus, money did not grow on the palm trees. (One suspects there is a research project underway to make this happen, though). Forsythe was persistently troubled by the University's lack of generosity (322). Yet other resources were found: necessity is still the mother of invention. As with mothers in general, some are more nurturing than others, and some seem quite gifted in their production of prodigies. Such was the case here. The relentless pressure to find extra-university sources of income pushed the professors to diversify their academic affiliations and intellectual interests and to cultivate corporate and governmental ties. As was consistent with his philosophy of "steeple of excellence", low priority was placed on undergraduate studies, and "obscure" areas of study were typically discounted.

Forsythe undertook self-conscious managerial policies for the cultivation of the very smart, or as he called it, the cultivation of selected 'spurs of excellence' in a few areas (spurs on the steeples?). These included AI, numerical analysis, and systems. The research-oriented plan was supplemented with an aggressive administrative approach. Traditionally, the faculty of a university is paid with monies coming from tuition paid by students. This is called being "on the gold standard" (323). The Terman style of leadership encouraged procuring a larger number of appointments by relying on "soft money" grants from the government and

foundations. Forsythe worked on this principle and relied on joint projects with hard science departments, which are chronically well-funded (e.g., medical and other parts of engineering school). He further leveraged the faculty with ‘courtesy appointments’, that is, professors from other departments who also taught in Computer Science. Finally, he was crucial in procuring important hires for various different institutes throughout the vast university. These in turn were encouraged to be self-sufficient. Feigenbaum emphasizes the implementation of Terman’s phrase, “every tub on its own bottom” (324). As a result, Stanford’s Computation Center, the Gonawanda from which the other continents of computing at Stanford broke off, was augmented by numerous other institutions.

Many of Forsythe’s choices were people with a clear acumen for leadership (325). For instance, William Miller, trained as a physicist, who Forsythe recruited from Argonne National Laboratories to run the SLAC computing facility and to be one of the first professors, later became chairman of Varian Corporation and Borland International, as well as Provost of Stanford. However, the intellectual ecosystem, particularly at the beginning of a discipline, may well need to be filled with diverse niches. Stanford’s salient variegation of the flora and fauna of academic life surely is part of the explanation for the system’s success. The functional differentiation of roles is notable for any denizen of Stanford, or similar universities.

One beneficiary of this ecumenicism was John McCarthy, who received his own research institute, monies, and students, but apparently was not burdened with requirements that he engage in the drudgery of institution-building. When the MIT administration stalled on implementing timesharing, McCarthy took up Forsythe’s recruitment offer, moved to Stanford in 1961, and established the Stanford Artificial Intelligence Laboratory (SAIL). McCarthy conceded that he was never an enthusiastic grantsman or administrator (326). He asked for money only from ARPA, just as he had applied only to CalTech for college admission. When he established the SAIL, all he did was to announce that it existed (CBI interview; 327), with neither press release, nor public relations campaign, nor damage control, nor buzz, nor spin. SAIL proved to be remarkably productive, as we shall see soon.

Edward Feigenbaum, and a host of others, followed McCarthy's arrival. Feigenbaum had moved to the University of California at Berkeley's School of Business Administration upon his graduation from Carnegie Tech, but found the nascent political unrest distracting and the environment incompatible with his interests (328). He also confessed to being no longer fully engaged with EPAM (329). Feigenbaum started at Stanford on the first day of 1965, and promptly located the center of the administrative, grants-finding, and scientific action. Forsythe brought him into managerial roles almost from the first weeks of his arrival. Feigenbaum also established the Dendral Project, which was closely linked to research done at Carnegie Tech and which was thus discussed in the last chapter- within the year. Forsythe conceded the directorship of the Computation Center to Feigenbaum, who raided Berkeley- fair game- and brought in systems engineers to construct computers for the Stanford Linear Accelerator, and Gio Wiederhold, a professor with expertise in database programming.

Finally, Feigenbaum and Forsythe built up computing facilities campus-wide, rather than only running a research department for graduate students. By deliberately focusing on the Campus-wide facilities, they were able to find powerful collaborators, such as Nobel winner Joshua Lederberg to help them obtain support from diverse funding sources (330).

The growing core of founders of computing at Stanford moved on their own initiative. But they were also encouraged and given monies by sources outside their own university. J.C.R. Licklider traveled among universities searching for promising junior professors to offer ARPA research grants, and in this process approached Feigenbaum at U.C. Berkeley. Licklider also sought John McCarthy's participation as a principal investigator. His knowledge of these two young people was attained through conversations with Newell and Simon, and by personal acquaintance with McCarthy, respectively (331). Feigenbaum readily concedes ARPA's steady signature hand on the checkbook, but he maintains that the technical content of the work done was "all ours- 100%". Prior to 1970, very little in the way of formal proposals was needed. Feigenbaum characterizes the proposals as "little more than a

handshake". These were unsecured loans, with practically nothing but the researcher's renown and word as collateral. However, the recipients were extremely creditworthy.

3. SAIL and the 'Golden Age of AI at Stanford'

The Foundation of SAIL

The Stanford Artificial Intelligence Project was established soon after John McCarthy arrived, and won ARPA support the next year. With relatively little ado, McCarthy received further, much more extensive, grant support two years later. This allowed the lab to obtain a large time-shared system, and to begin working on vision and robotics; this in turn was followed by a superior PDP-6 computer and better facilities (332). The Laboratory was the original location for the Dendral Project, which formalized scientific knowledge and provided the basis for expert systems. The latter has often received a great deal of attention, certainly in part because of Edward Feigenbaum's energetic public persona, and because expert systems became a popular topic outside AI itself.

Raj Reddy calls the late 1960s and early 1970s 'the Golden Age of AI at Stanford', crowned with numerous achievements. Formalization of scientific knowledge, and subsequent fielded production systems, are only one among numerous such achievements. Others include much instantiation of receptors and effectors of physiological processes, such as haptics, vision, and robotics. Both planning and anti-planning, as we might call the school of 'intelligence without representation' pioneered by Rodney Brooks, emerged from the SAIL and the closely affiliated laboratory at SRI. SAIL was the site of natural language understanding projects, started by Roger Schank and Chris Riesbeck before they established a large concern in this area at Yale. John McCarthy and others proceeded with computer language development, formal languages, and epistemology. Others worked on hacking of systems and environments, created some of the earliest computer music and computer games and utilities such as news-services readers, which were immensely sophisticated for their time. SAIL was the basis for much original work in AI, carried out by many people. However, the Lab did start with one person, and we shall digress

slightly to consider one of the scientific world's most interesting individuals.

The Ongoing Singularity of John McCarthy

The late John McCarthy was perhaps the world's most perfect paragon of the computer nerd- one might say, paraphrasing Gilbert and Sullivan, the very model of a model computer nerd. He epitomized every one of the stereotypical traits of such individuals, but like many nerds, actually participates and in some ways excels in social and athletic activity in a rather unique way. He clearly lacked interest in small talk. This was often seen as shyness, but as Pamela McCorduck tells us in Machines Who Think:

'his long silences in ordinary discourse... [are] because he can't think of anything to say that's worth saying...' (333).

Any impression of genuine lack of involvement in the human and intellectual world was contradicted by McCarthy's leadership in university and academic life, including voluminous correspondence with many people, over fifty years. The issues to which he devoted himself historically seem to indicate a relative lack of concern with moral relativism or foibles, that is, the 'human' side of issues. However, the other side of the lack of engagement with human and university politics is a appealing absence of artifice or malice. McCarthy's sense of justice, impartiality, fairness, and overall decency were widely known.

McCarthy did everything his own way, from often cantankerous personal mannerisms to reported acerbic commentary on Stanford's computer science faculty email list. McCarthy's choice of sports was unconventional and even bold. He took up mountain climbing despite a marked lack of physical coordination. (His second wife, IBM programmer and climber Vera Watson, died while ascending Annapurna in 1981 (334). On his World Wide Web homepage, McCarthy published the longitudinal and latitudinal coordinates for his office and home, with road directions included for those who eschewed the use of compasses.

McCarthy left Cambridge in 1961 and settled in at Stanford. He became involved in the cultural vagaries of the 1960s in various ways, sometimes even inadvertently. McCarthy's ties to Cambridge were not severed as smoothly as they might have been. In preparation for the cross-country move, McCarthy sold his house. He extended a second mortgage to the purchasers, two Harvard psychology professors, whose names were, alas, Timothy Leary and Gordon Alpert. The two established their first hippie commune in McCarthy's former house. Not long after the sale, they were fired from Harvard because they gave LSD and psychedelic mushrooms to their students (335).

McCarthy also experimented with both the political and the computer radicals of the time. He had been a red-diaper baby, as those raised in the Old American Left are called, but had quit the Left in 1953. The New Left turned him sour, though (336). He joined the Mid-Penninsula Free University, which offered teach-ins on various worthy subjects, and helped them to obtain a five thousand dollars grant. The first meeting, in Menlo Park, was disrupted by a group of 'ultraradicals', who drowned out the other speakers and advocated killing policemen. He visited the USSR and Czechoslovakia just before 1968 Soviet invasion. The visit and the subsequent invasion, combined with experiences with the more strident forms of the American New Left, led him to leery of radical leftism. Always the contrarian, he swung to the opposite extreme and began wearing suits. McCarthy had his famous dense shock of hair cut short, and then went one astonishing, breathtaking step beyond even this:

" I even acquired something of an affection for Richard Nixon !" (337).

McCarthy's wider intellectual peregrinations deserve mention. These took the form of statesmanship, engagement in non-partisan current political issues, and efforts to widen access to computing facilities. McCarthy's traditional abstinence from university, science, and party politics did not mean that he did not engage in statesmanship. He and other members of the Committee of Concerned Scientists wrote letters to the authorities on behalf of the jailed Soviet computer scientist and agitator on behalf of religious freedom Natan Sharansky in the mid-1970s (338). He

considered rights to privacy and freedom of expression of information stored or transmitted by computer, and wrote a “bill of rights regarding computer-kept information (339). McCarthy also participated in the National Legal Center for the Public Interest (340). Other issues of political concern he wrote about were central to their time, but otherwise brainy and non-partisan, such as “Energy for the ‘80s” (341), Project Independence, ‘a project aimed at making the U.S. substantially independent of foreign energy sources, 1974 (342), and extensive materials on the topic of Ecology (343).

First, finally, and foremost, McCarthy worked. He obtained research money from ARPA. His approach to office space was expeditious and low-budget. In 1966, he discovered the Donald C. Power Laboratory on Arastradero Road, in the hills above Palo Alto, several miles west of the main campus. General Telephone and Electronics had constructed and then abandoned this building, on Stanford land, some years earlier. The university let McCarthy and friends occupy it for over a decade. The Powers building was not maintained, and had to be torn down in the late 1970s (344). At that point, McCarthy and the others moved to Margaret Jacks Hall, on the main campus. SAIL itself was formally shut down in 1980, as McCarthy did not wish to take on the managerial responsibilities of running a laboratory (345).

SAIL as an institution, despite its foundation by a person who would seem anathema to the very idea of running one, was immensely fruitful. However, one is tempted to say that in its early days at least, it does not resemble its founder as much as the other two major centers for AI bear the stamp of their respective leaders. Was the SAIL a research program in the ‘strong’ sense of intellectual coherence, as the CMU program was for a good number of years, or was SAIL simply a gathering of extremely smart people in one convenient building ? Knowing even the first thing about McCarthy, and the second thing about the work done at SAIL at this time, the ‘people in a building’ image seems to fit better (346).

Still, the whole was far more than the sum of its parts: one sees a collective virtuous circle of intellectual energy. It was relatively undermanaged by the standards of university or grants

administrators (even for the late 1960s), but once an administrator was hired to elicit progress reports, the requirements of reporting were met (347). More generally speaking regarding computing at Stanford at the time, the light-handed administration of research by ARPA was successful. Even when it did not produce the required reports, research did produce computer science, and this was the bottom line of output that kept the place flush with cash. Moreover, during this time an open-ended approach to this sort of work prevailed, as Raj Reddy described the period:

“...there was, in his [Feigenbaum’s] mind, at least, no necessity to worry about justification of these projects. You had the money; if some interesting problem came along you investigated that problem until you were satisfied you had a solution or the problem wasn't very interesting anymore.” (348).

But despite the general tone of easy money, by the late 1960s people at SAIL were writing reports to ARPA (349). During the “golden period of AI at Stanford”, the laboratory worked in at least six major fields, and inspired other things as well. In the central several years of the 1960s, the laboratory worked in emulations of human psychology; understanding speech; heuristic programming (that is, the origins of the Dendral Project in 1964 and 1965); chess-playing programs; the opening up, but not the closing of, the Frame Problem; robotics both in the sense of robots that moved themselves around and in the sense of robotic arms integrated with vision; machine vision closely attached to robotics; the ongoing work of McCarthy and the hackers to inaugurate timesharing and systems capacities for the PDP; LISP and other formal machine languages; and endless minor hacks to make life at SAIL more pleasant and full of conveniences for the local residents. Other things, for instance the early work of Stanford professor John Chowning in computer-generated music, were immensely inspired by the residence of their creator at the project (350).

Robotics in the 1970s

If McCarthy was interested in robotics from the early days of his life on the West Coast, so were quite a few other people. This inventory includes involvement with robotics in several senses of the term. By

definition, early robots were ponderous bodies which had to learn to navigate a big world, however simple and schematic that world may have been. (Any body which is problematic to move, must be referred to as ponderous). At SAIL, these were developed by Hans Moravec, whose illustrious career in robotics continued, and continues, at Carnegie-Mellon. His robot was prosaic, not anthropomorphic. It was a computer-controlled cart topped with a TV camera mounted on top, "which steered itself around the redwood deck and the road circling the lab" (351). Another sort of robotics was the design of flexible arms and hands. Jerome Feldman and Victor Scheinman designed robotic hands and arms, known as the PUMA, which were widely adapted by industry (352). Helpful to both of these sorts of robotics was extensive research in computer vision, which at that point included the most basic distinctions such as edge detection.

The Frame Problem

" There is a time in science where it's too early to attack a problem. The stuff has to bubble a bit more and you have to be able to have the foundation on which to stand." (353).

The experiments in problem-solving and search initially explored by Newell, Shaw, and Simon had provided their own knowledge bases, however finite. But when the input to the system is the 'real world', in which knowledge is more fickle and the nature of information more heterogeneous than in 'toy worlds', how is that knowledge to be maintained ? Bitting off a piece of reality that is large enough to be tasty but small enough that it won't get stuck in the throat is a difficult call.

The issue of narrowing the database for problem-solving was confronted by Patrick Hayes and John McCarthy in "Some Philosophical Problems from the Standpoint of Artificial Intelligence" (1969). Their paper introduces the Frame problem, as it is known. The frame problem was, of course, brought up by Hubert Dreyfus: he proposed that in principle and in practice, the problem was insoluble. Hayes and McCarthy proposed ways to begin to solve it.

AI thus far had limited its chosen problems to contrived 'toy world' data, they said. But in order to achieve intelligence that bore more verisimilitude to human ability, AI programs must maintain more heterogeneous data, about more things, and must display the ability to analyze things commonsensically. Exercising 'common sense' in order to separate the wheat from the chaff is perhaps the gist of being human: the frame problem is the recurrent challenge of where to draw boundaries around segments of a problem, such that those segments are computable.

Hayes and McCarthy illustrated the frame problem by showing just how much one must know in order to simply find a telephone number. This is usually a mundane matter, but when the requisite number is missing from its designated place in the phone book because it has been inked out, or because the page was torn out, the problem may be a challenge for algorithmic solutions. The very large number of caveats that may be applied to even something this simple suggests that fully stating the necessary formalisms is almost impossible- for one might have to specify that the minutiae even for simple [sic] circumstances are too great to enumerate. Hayes and McCarthy show how the sheer volume of data that might be pertinent to solving a commonsensical problem greatly hindered the solution of such problems. They introduce situations, "complete states of the universe at an instant of time", and fluents, particular attributes of situations (e.g., raining. They also introduce modal operators, that is, logical expressions that limit the occurrence of any given fact based on previous circumstances, into AI (354). Modal operators and the associated 'situation calculus', as it is known, depict facts about the world at moments in time, and allow for the expression of alternative consequences of different events or choices.

It is better to pre-empt one's critics than to ignore them, and AI had actually confronted the frame problem, as it is called, several years before What Computers Can't Do appeared. The issue had, moreover, been considered less obviously, in programs such as Pandemonium, which sought to emulate the chaotic perceptual field of real living things. Production systems, in a far more closely channeled way, were influenced by Pandemonium's design with parallel processing. In both of these cases, there was an evident

tradeoff between the depth of knowledge about any given field being considered versus the heterogeneity of knowledge sources. In the early search and problem-solving experiments, the preference was always given to a singular space of knowledge, with the simplest kind of machine input. But other AI scholars were puzzled as to the means to limit search when the inputs were the immensely diverse inputs given to a human being, and the search space was, potentially, the human's practically infinite stock of data. These techniques have been greatly enhanced since then. The perceived need to engineer a computer program that thinks commonsensically has engendered a subfield of computing and philosophy called naive physics, which expresses knowledge about the stock of informal knowledge about objects available to all people of normal intelligence (e.g., raw eggs have brittle shells; water does not flow uphill; if you jump in a lake you will get wet).

"He [John McCarthy] proposed a technique for dealing with it called "circumscription". Since then there have been a whole host of people in AI who have looked at the whole problem of what is called "non-monotonic reasoning". Circumscription is one approach. Ray Reiter, who is now at Toronto, has proposed some techniques called default theories. These techniques now can be incorporated in planning systems to help us deal better with the frame problem. It doesn't really solve it, but [we are] much more sophisticated than we were in 1971... (355).

The Frame Problem was posed in 1969 as a strictly intellectual quandary. But it might as well be understood as a double entendre with a political side. McCarthy and Hayes were concerned with the syntactic and semantic nature of knowledge representation. They could also have been thinking as well about how to define a discipline so that its questions are crisp. The postulated course of events for the development of a science is that it be 'normalized. Both politically and methodologically, said maturation of a scientific research program entails (term used colloquially) a narrowing of interests. This is often needed simply in order to 'get one's hands around the object'. Doing this, even in terms of computational formalisms, was obviously very difficult. McCarthy's fundamental orientation has apparently always been with the protocol for expression rather than with knowledge representation. But the frame problem is not solely computational. It may not be fundamentally solvable at all, despite technical fixes. The problem

posed as the Frame Problem is simply how to circumscribe (this word is used colloquially) the computational formalism of any given issue. This is problematic because of the contingent contexts which may alter the solution of those problems. That is, problems are situated, and the abstractions which sufficed to describe the World in the toy worlds scenarios in which AI lived in the 1960s could not suffice with a world of real knowledge representation. Once one steps out of the toy worlds, this is invariably an issue. The contingencies, many of them semantically trivial, are intrinsic to any real world problem. The real world is not a closed world.

The Frame Problem addressed the hangnail which Dreyfus brothers' had found. They had claimed that computer programs could not have fringe vision, or 'focus' on a given problem. The frame problem poses the same concern. ostensibly, a system without intentionality, or, roughly, a 'self', cannot engage in search which selects a proper goal state. The assertion that computer programs cannot 'concentrate' is on its face unacceptable because the criteria for 'concentration' is phenomenological rather than behavioral. The essay of intelligent systems has historically set itself to a behavioral criterion of intelligent output and intelligent information processing, not to an intangible internal state test. It would seem that McCarthy and Hayes redefine the intentionality issue as a computational rather than a semantic or philosophical problem. (The issue of intentionality will be addressed in the first part of Chapter Nine). The substance of the critique which Patrick Hayes (then at Edinburgh) and McCarthy (at Stanford) proposed to the AI community was a constructive version, perhaps, of the entirely negative assertions made by the Dreyfuses.

The latter stated that consciousness was holistic in all regards and therefore could not be studied piecemeal at all. Hayes and McCarthy recognized the effective multiplicity of connections between every bit of knowledge and every other in the actual human mind. This did not preclude the possibility of artificial intelligence but simply recognized how difficult it actually was. Their proposal of this issue begged the question of the establishment of a wide, multi-faceted semantic representation of human knowledge. This was not addressed until the Cyc project in the mid-1980s and knowledge sharing and ontological engineering development,

several years later. References to the Frame Problem include the original Hayes and McCarthy statement, "Some Philosophical Problems from the Standpoint of Artificial Intelligence", and The Robot's Dilemma, a compendium edited by Zenon Pylyshyn (1987). In another sense, everything is a reference to the Frame problem. Efficient search itself implies that the frame problem, or the threat of computational explosion caused by too many inputs to a state space, has been thwarted (356).

4. Utilities and Software Applications

The formal academic research as discussed above, of the sort that led to research papers and careers as professors, was one product of the academic work conducted at the laboratory environment. Other forms of computer work were conducted on the laboratory environment. By this we mean the development, first of all, of better a computer environment and systems at the SAIL (hardware, programming environments, etc.). Second, we refer to the amenities and utilities, both computational, such as computer games and wire service reports, and strictly convenience-related such as computer-controlled food vending machines. Computerized consumer goods are ubiquitous circa the turn of the century: but they were not in the 1960s, and these were important innovations, even if they were side products to the "main" activities at the SAIL. Thus developments that we could see as incidental were actually one of the bigger genres of contributions.

The SAIL's cornucopia was doubtless a part of the very enfolding nature of the facility: it was what we would now call a '24-7' environment. It is not clear that the wording "24-7" existed at the time. In those days, one instead said, quaintly, that people were at the facility at all hours of the day and night. Life at the office was life in the office, both night and day, in an early version of the frenzied workplaces common in Silicon Valley (357). Despite the introduction of Les Earnst as manager, at ARPA's behest, to keep things sane, he ran the hotel better rather than kicking out the residents. The building became a legendary haunt of hackers and tinkerers. Roboticians Rodney Brooks and Hans Moravec shared an office there during graduate school. Moravec is reputed to have lived in the rafters of the building, as did other people (358).

The computing environment at Stanford when McCarthy arrived did not even involve timesharing. This promptly changed, and with it the availability of computers to students. With the help of J.C.R. Licklider, McCarthy replaced an IBM 650 and a Burroughs 220 computer with more modern machines by the same manufacturers in 1963. DEC gave him a PDP-1, the new machine with a CRT display capability. A couple of years later, ARPA began to foot the bill for vision and robotics research at SAIL, and paid for the replacement of this machine with a PDP-6, a much heavier and more interactive timeshared system. This was one of the first such machines installed, and for its time it was excellent:

“ One of DEC’s first PDP-6 computers was installed at the Lab with 64K of core memory, and a timesharing monitor. Long-term storage was on DEC tape and terminal interaction was through Model 33 teletypes. Very soon after, six III display terminals arrived and by 1971 Database displays were in every office.” (359). This was a highly advanced system, apparently “the first display-based timesharing system after Philco delivered twelve [displays] in response to Stanford’s specifications.” (360).

The PDP-6 was so revered an artifact that even in its dotage many years later it was given a funeral and a decent burial. Raj Reddy told a CBI interviewer in 1989:

“ SAIL, Stanford AI Lab computer, the PDP-6, PDP-10, is being disconnected this month... this week actually, after 25 years or something...They are actually having a closing-down ceremony for that computer that was actually being there and operational, and just now, I think, it became too expensive to keep it maintained” (361).

At the same time the laboratory moved to the Powers building in the hills. The boost in computational juice, timesharing, so that more people could nestle into their terminal cubicles, and a room with a great view, allowed wider student and researcher access to machinery, and helped boost the plenitude of projects. The resources may have been relatively limited by current technology, but this does not mean that people’s visions were: “ In those days

McCarthy is reported to have thoughts of a PDP-1 flying a small airplane with optical feedback from a TV camera.” (362).

They say that the rich have more money, but in this case, money was not particularly at stake. The rich had and have better computers. In addition to research per se, the availability of computing resources allowed countless systems hacks, and hundreds of other things. Projects being pursued circa the early 1970s included system and infrastructure improvements in the form of better languages (SAIL; MLISP2; LISP70), and more efficient timesharing systems (363). Applications improvements at SAIL improved the rigor of technical and engineering functions. During the early 1970s, for example, the SAIL introduced ‘PUB,’ a text editor for technical papers. The editor could take care of a number of the minutiae of editing (364). Other projects augmented engineering work, such as the interactive design system for digital logic, which allowed the user to specify the desired logic elements, and model and test them in software rather than in material demonstrations (365). Both of these were so well-received that they were adopted at CMU and several other institutions.

The SAIL was also one of the early inventors of an online newsreader. Newsreaders are old hat today, but reading news as the AP wire service published it was novel at the time. The program, APE, was an editor which allowed the choice of stories using combinations of keywords (366).

Another program similarly embodied rules and heuristics for a more highly algorithmic domain, specifically that of the formation of polyhedral models. Geomed, which found applications in animation, mechanical design and computer vision, could facilitate the construction online of objects which were true to geometric principles (angles adding up properly, forms not intersecting, etc.), when commands were typed in, in LISP or Algol (367). SAIL also contributed to the ARPAnet, in the form of work on the improvement of protocols for internetworking. In the mid-1970s, ARPAnet was being tentatively taught to talk with other network projects, such as TYMNET, networks at the University of London-NPL, and the French project Cyclads (at the IRIA research center). Networks outside of one’s own host or domain name are currently

accessed with relative ease due to intense work on host-host communication, over a number of years, but this is the sort of fundamental work to which SAIL contributed (368).

Other inventions were pure whimsical hacks, the confetti of the computer world. If a technical editor which would automatically format documents to look respectable was useful, a word processing program which would print in various scripts and sizes, including Elvish (used by the characters in Middle Earth), was useful in a different way (369). A radio-controlled channel selector on the TV, and a computerized accounting system that ran the vending machine by personal account charges rather than by coins, were useful as well (370). SpaceWar, the computer game that McCarthy's prime hacker Steve (Slug) Russell invented and brought with him from MIT, was upgraded from the PDP-1 to the PDP-6 "as one of the necessary programs for the new machine" (371). Computer gaming continued, predictably with more references to Tolkein (372). Core War, another computer game for two adversarial players, was invented at the SAIL (373).

The SAIL even contributed to culture: it introduced a word that is second only to 'SNAFU' in expressiveness and pungency, or 'live long and prosper' in perspicacity. Foobar, or dubar: f----- up beyond all recognition; foo' is to lightly deprecate anything or person.

A different area of the Stanford University computing gestalten was actually not at Stanford, or at least not preponderantly at Stanford. McCarthy was a distinguished contributor to the "radical computing movement", as the author will call it for want of a better title, and perhaps this is the place to mention it. Computing done had historically been done in an Adam Smith economy, that is, in an environment in which processing power and computing facilities were so poor and so limited that only the military and select commercial enterprises received anything. (AI was exempt from this rationing because of its good graces with the powers that be). But the "Computer Left" envisaged something far beyond this: plenty of computing power for all. With enough technical advances, this vision came true. As every one of the cavalcade of advances- the IC, timesharing, CRT screens, teletype input, the light pen, the mouse, online editing, VLSI, the microchip, etc.- took place,

adherents of radical computing attempted to rush such tools to the general population as fast as possible. In helping the Homebrew Club, and in affiliating himself with early pop computing organizations like the Bay Area Home Terminal Club (374), McCarthy represented the technical vanguard of this movement.

5. The Stanford Research Institute

The fourth major center for early AI research was not a university at all, but a think tank as large as many colleges, originally attached to Stanford. Stanford's administrators had first discussed the idea of a think tank in 1939, but they could implement the idea only after the War ended. The Stanford Research Institute, now SRI International, was established in 1946 as a wholly-owned subsidiary of Stanford University. SRI was expressly intended to pursue research which would further the industries of the Pacific- to "encourage regional economic growth by undertaking commercial contract research too specialized, expensive or speculative for industry alone"- as well as to promote the research needs of Stanford University (375). The Institute was established in nearby Menlo Park in what had been a large military hospital, used to receive the wounded from the Pacific during the Second World War (376). Despite the cushy wealth of the place, the original buildings were not fancy. Engelbart does not mince words, calling SRI's first facility "a low-rent, initial facility", with many buildings of "one story, quite sleazily built' (377).

It is wise to bear in mind the relative novelty of such an institution. Dickson's Think Tanks describes the anemic academic capacity of the American West, even California, in the immediate PostWar years:

" In 1946 R&D was concentrated on the East Coast... in 1946 the eleven westernmost states had fewer industrial research outfits among them than the state of Connecticut." (378).

Solicitations to attract corporate contributions reached far outside the Stanford environs. Fundraising efforts for SRI called upon corporations more broadly identified with the "Pacific", apparently meaning all of coastal California. David Sarnoff, chairman of the RCA Record Company, helped to raise more than one million dollars

in 1950 alone (ibid.). The local industry focus and the aggressive fundraising worked: by 1970 revenues totaled about sixty million dollars per year, and the total staff numbered about three thousand. The agenda of “Pacific” research devolved into one of Federally-sponsored research, synergistic to the capacities of state industries such as electronics, aerospace, and avionics. By the start of the 1960s, three-quarters of SRI’s revenues came from mostly military Federal contracts (379), in general congruence with the basis of California’s postwar prosperity.

SRI was (and remains) also exceptionally diverse, versatile, and run with an eye to new opportunities for contract research. According to various accounts, during the major part of the 1960s, higher administrative oversight was chiefly concerned with ongoing financial backing for projects. The technical visionary Douglas Engelbart and the science journalist Dickson both emphasize this quality in SRI’s management. Extreme opportunism had mixed results, which included both the nurturing of Engelbart’s invention of the computer mouse and PC working environment, and the virtual expulsion of Engelbart from SRI when he proved too independent and resistant to suggestions- at least from Engelbart’s own perspective (380).

In other areas, SRI management has excelled at innovative research quality, and at cost overruns. SRI’s work has historically been a grab bag of every sort of topic which people were willing to pay for: studies of underwater communication among sea lions, the gambling habits of casino regulars, many significant advances in information technology and AI, and electronic ambush aids for the Vietnam War. The latter contributions to modern warfare would lead Stanford University to formally divest itself of SRI in 1970, but several of the information technology advances are among the ‘greatest hits’ of PostWar computing. These include the odd typeface used in financial documents, the computer ‘mouse’, and early AI research on robotics and planning.

A perennial and ubiquitous contributor to computer science, SRI was involved in computing technology even before Stanford was. Perhaps this was because business was interested in military and commercial computing during the period when the field was

specialized and highly capital-intensive. The topic was actually handed to SRI by one of its clients. As early as 1950, the Bank of America (now BOFA, or BankAmerica) wished to establish a machine-readable language for the automation of financial documents. The Bank commissioned SRI to develop a machine-readable typeface, in a secret project ERM called the electronic recording machine (381). The name was later modified to ERMA to make it slightly more palatable. Accordingly, Kenneth Eldredge of SRI invented MICR (magnetic ink character recognition) in 1954, and the type was used on the IBM 702, one of the very first civilian commercial computers (382). Experiments proved electronic check processing feasible, and the banking industry as a whole took on ERMA as its standard. The machine-readable type was adopted, with slight modifications, as the inescapable font on the bottom of checkbooks and other such documents. The ERMA project was the initial core of SRI's Computer Science Laboratory, and brought in the first of what became a large and diverse group of computer scientists to SRI, and Joseph Weizenbaum, who later became an MIT professor and critic of the excesses of AI and computing more generally, began with ERMA. So did Louis Fein, the consultant who wrote a report which helped to inspire Stanford's administration to establish computing as a field of study.

Thus in the mid-1950s, SRI hired its first ersatz computer scientist. But computer scientists were made, not born, and were not even quite aware that they were entering a new profession in itself. The Institute was highly tolerant of discipline-hopping on the part of its researchers- a perspicacious policy. Charles Rosen entered as a physicist and emerged several years later as a computer scientist who built 'self-organizing systems' and 'learning machines'. Rosen and the people who gathered around him were inspired by Rosenblatt's celebrated Perceptron, and proceeded to build one of their own in 1961. The Applied Physics division sponsored this work, and establishing a Learning Machines group. The Minos I and II were fundamentally pattern recognition computers, which used local 'neural' modeled threshold firing rules to detect patterns. The relations between threshold elements was altered in various ways by motor-driven potentiometers, and the machine's neurons would respond with various values, which converged as a function of statistical averaging. Rosen, A.E. Brain and George Forsen

developed MINOS I, a trainable pattern recognition device, by 1961. By the time Richard Duda joined the group in 1963, the team was engrossed in building Minos II with money from the Office of Naval Research. In 1961 Nils Nilsson, another engineer who joined SRI after earning his Ph.D. from Stanford, began work on decision rules and training procedures for use in collecting by computer sample statistics on observed patterns, which used Minos I, a perceptron built by SRI, and resulted in his 1965 publication Learning Machines (383). The Minos projects were at least as much about statistical simulations of 'learning' as they were about cogitation, but they were effective in the area of pattern recognition. The work continued under various military sponsors for several years, culminating in a project for the Signal Corps, which combined rudimentary machine vision with recognition of hand-printed FORTRAN code (384).

This work thus had no "familial" connection to symbol processing digital computers. In fact, some of the members of the SRI contingent eventually learned about 'Classical AI' only in the mid-1960s when they read Marvin Minsky's "Steps toward Artificial Intelligence" in Computers and Thought (385). Minsky himself worked with the SRI learning machines group in 1964, as they prepared a proposal to construct a robot with both pattern recognition and heuristic computing capabilities. Bertram Raphael, one of Minsky's doctoral students, arrived at SRI after graduation and a stint at Berkeley, and began teaching the group LISP. Raphael and Nilsson were also affiliated with the SAIL, which helps as well to account for SRI's increased engagement with 'classical' AI after 1961.

The Genesis of Shakey the Robot

Robotics developed naturally from this research on Perceptron-type local learning networks. The idea of intelligence as a physiological phenomenon appears to have led to trying to implement more anthropomorphic receptors and effectors. As we saw earlier, robotics at SRI emerged from the further adventures of physicist turned computer scientist Charles Rosen. His first conception of a robot was as a means to vary the type and complexity of input and output and thus expand the scope of his learning machines work:

“...satisfied that MINOS II was nearing completion, Charlie became absorbed by a more grandiose vision: ‘ What would it be like ...to build a large learning machine whose inputs would come from television cameras and other sensors and whose outputs would drive effector motors to carry the machine purposefully through its environment.’ ” (386).

Rosen wrote a memo describing the machine as a “robot” or “automaton”, and ARPA endorsed the proposal late in 1964. They did not grant money to build this machine for art’s sake alone: the research was specifically intended to develop “automatons capable of gathering, processing, and transmitting information in a hostile environment” (387). In addition to ARPA/ DARPA, the Information Systems branch of the Office of Naval Research also supported SHAKEY at various stages of its development.

This was the origin of SHAKEY the robot, which spawned and was joined by other successful projects in machine vision, effectors, natural language understanding, and QA4 and STRIPS, two of the earliest planning languages (388). The attempt to build such an automaton engaged various problems in AI: robot locomotion, vision, haptics, navigation and planning, and in the final incarnation of the project, natural language understanding as well. This is the sort of study that establishes a way for other things, by dint of necessity. Rather, the need was perhaps more exigent: it demands their construction.

We have already noted that tropism toward light was one of the guiding principles for some of the very simple robots which ‘lived’ elsewhere at the time. The insect brains of the robot at the NRL and of the later generation of insect robots built by Rodney Brooks, Patti Maes and Maja Mataric, were indeed capable of apparently intelligent movement because their sole commandments were to move toward or away from designated stimuli. But Shakey was intended to have greater navigational capacities and greater semantic understanding. Thus, it needed more than an insect brain. The early work on the robot focused more on the representation of the floor space than on receptors and effectors:

“ Much of our early work with Shakey was directed at studying navigational algorithms for calculating routes across floor space cluttered with various

obstacles. We explored several techniques for representing key points in space as nodes in a graph, so that graph searching methods could be used for route planning. Out of these studies, Peter, Bert and I developed the A* algorithm with a heuristic component in its evaluation function. (For robot navigation problems, the heuristic component was set equal to the straight-line distance to the goal location).” (389).

The idea of understanding the terrain to be navigated, and taking high level commands and turning them into plans, meant new computer languages and the representation of plans. A number of SRI scientists, including Waldinger, Derksen, Rulifson, Nilsson, Raphael, and Hart developed QA4, a language for robot plan generation. In QA4, ‘a plan is offered as a list of actions,’ and thus locomotion problems, which the creators described as ‘absolutely trivial’ could be addressed by a hierarchical breaking of larger plans into series of much smaller actions (390). The early Shakey the Robot, developing over the course of the late 1960s and early 1970s, developed an intellectual world, albeit hardly one which was commensurate with a soaring intellect. The entire vocabulary of the Shakey world could be described in a list of about ten verbs (391). The physical nature of the Shakey world was a grid of the Cartesian coordinates, a scheme which was easily rendered digital. Lest we say that this is too simple and schematic, we should note that the system had a ‘demon’, a technique for coping with surprises such as doors opening. If a surprise took place, the system backtracked, handing the process to the parent node of the process that was being executed at the time (ibid.).

Q4 was accompanied by STRIPS and ABSTRIPS, a sequence of planning languages invented at SRI by the robot group, joined by Richard Fikes. The Shakey the Robot sequence consisted of a robot running using a planning language. STRIPS, written in Planner, was the control procedure for Shakey the Robot. Shakey added further impediment to its tasks in that it had an ersatz form of ‘self-consciousness’. It constantly had to evaluate its own state location, although this is probably more technically difficult as far as robotics is concerned than analytically difficult for cognitive emulation concerns.

In addition to advancing robotics and multi-modal AI fieldwork, Shakey- Strips was also an erudite commentary on and advance

relative to the General Problem Solver. The historical roots of planning languages are generally depicted as being in the difference tables found in GPS. Richard Fikes, a major Shakey-Strips contributor, earned his Ph.D. at CMU. GPS listed as predicates the state of world and the goal state. Planning languages did this too but were somewhat more applied. The difference between the current state and the unmet goal state is enumerated as component predicates not yet matched by the (current) state. Like the later GPS versions, planning languages create macro-operators incrementally. They agglomerate intermediate goals by altering the add and delete portions of lists, that is, the control structure. In place of a difference table which listed ways to reduce the differences, a planning language relies on the simpler means of adding and deleting from the goal state list; as the add list and the delete list. (As explained earlier in the chapter, means-ends analysis works through analysis of discrepancy between goal state and current state). Strips/ SHAKEY communicated with the user in simplified natural language, although this is not where its novelty lies. Abstrips, a year later, was a further, more abstract excursion into planning languages; in Abstrips, there is more than one abstraction space (392).

Thus, in its prime, Shakey was something of a dummy by the standards of current robotics. But it had a rather impressive comprehensive roster of skills. It could take an instruction- it knew scores of words of written English- make a plan based on the instruction, work its way around limited foibles such as obstacles placed in its path, even cut apart knots, and cache its own answers. It used low-level vision- this phrase typically means that it had achieved edge detection and some degree of sorting out shades and shadows. This resulted in a performance that was impressive despite the slow and indeed 'shaky' quality of the movement. It impressed journalists, for instance Bruce Darrach, who profiled Shakey in Life Magazine in 1970:

"...The young scientist who was showing me through the SRI...sat down at an input terminal and typed out a terse instruction which was fed into Shakey's brain, a computer set up in a nearby room:

PUSH THE BLOCK OFF THE PLATFORM.

Something inside Shaky began to hum. A large glass prism shaped like a thick slice of pie and set in the middle of what passed for his face spun faster and faster till it dissolved into a glare. Then his superstructure made a slow 360 degree turn and his face leaned forward and seemed to be staring at the floor. As the hum rose to a whir, Shaky rolled slowly out of the room, rotated his superstructure again and turned left down the corridor at about 4 mph, still The robot's awkwardness, with its attached apparatus of TV camera, radio hookup, and cylindrical garbage can type construction, is legendary. Some of the criticism was on aesthetic grounds, which was far from what the designer had had in mind: Darrach says that: "It looked at first glance like a Good Humor [ice cream] wagon sadly in need of a spring paint job..." The machine became a media celebrity, insofar as AI programs can be media celebrities, with all of the usual caveats typically provided by science journalists who write about such things.

One can only wonder if such journalistic depictions had anything to do with the decision by DARPA to entirely eliminate funding to this project, in 1974 or 1975. Probably not, but the DARPA funds were indeed withdrawn, to the infuriation of all. In a 1989 interview, Nilsson attributes some of the problem with this project being funded to an ongoing political ambivalence as to the role of robots:

"...My perception was there were two feelings about this robotics stuff from the standpoint of DARPA and the military people. One feeling was [that] it's too far out. Maybe there will be robots that will help us on the battlefield, or will help us in supply depots, or warehouses in 2030, or past the year 2000. Maybe we should be working on these things, but it's too far out. We have to mobilize the technical community to deal more with some specific problems we have in this decade. But on the other hand I think they felt [that] maybe it was also a little too scary to be supporting. It was somewhat contradictory, because if it was so far away how could it be so scary? But, you know, "robots". We're telling our Congressmen we're building these robots that are going to do this, that and the other thing, and they want people to be in control. So they wanted us to orient the project much more toward what they called Command and Control Applications, in which a commander, given all the information would be aided by various kinds of decision aids, getting reports from the battlefield or whatever, ultimately would be making decisions, but would have certain suggestions presented to him." (394).

The project continued by stealth, essentially, in the form of the CBC Project, or Computer-Based Consultant. People involved with it immediately state that some subterfuge was involved in some of

the continued robotics work. The CBC was a virtual Shakey, minus the effectors, as Nilsson explains:

“...it involved using much of the same technology, software and everything else, except not for a robot... [it] had to do with building a computer system that could give advice to an apprentice technician about what that apprentice technician should do next in order to achieve a certain task: repair some equipment or something of that sort. From our point of view, instead of using the motors on Shakey the robot we were using the muscles of the apprentice technician. We were still thinking of ourselves as doing robotics. We were doing the reasoning, planning, and figuring out what should be done, and so on.” (395).

Douglas Engelbart and the Augmentation Research Center

SRI was also the point of origin for a separate development which was one of the keystones of personal computing, namely the NLS, or ONLine System created by the one and only Douglas Engelbart (1925-). The NLS appears to have been the ground under the feet of the creators of the Alto, at Xerox PARC a few years later, and is not entirely acknowledged as such. Other elements of Engelbart's creation, such as the computer mouse and the idea of an augmentative display screen environment, are more clearly accorded to him. He deserves mention in this work- and clearly much fuller study in other scholarship.

A Norwegian-American of working-class origin from Portland, Oregon, Engelbart was trained as a radio and radar technician, and was put into the Navy repairing these devices near the end of the Second World War. The story of his enlistment into the Navy is a tragicomedy of poor timing. His ship was leaving the Port of San Francisco on V-J Day (August 14, 1945) as the surrender of Japan was announced, bringing the entirety of World War II to an end. Despite the crowds cheering on land, the ship departed, and Engelbart spent many months in the Philippines (396).

While on a sojourn in the South Pacific, Engelbart read Vannevar Bush's 'As We May Think,' and found his calling in the idea of a device with a display screen, with knobs or levers as he initially put it, which would help people create written work. Engelbart characterizes himself as a 'naive drifter' during the early phase of

this development of his idea (Stanford oral history). This vision, successively refined as Engelbart proceeded through a technical post at the Ames laboratory, graduate school in EE at Berkeley (Ph.D. 1955), and the first years of a decade at SRI, placed him in family of the augmentative tradition, as we might call it. Engelbart was personally acquainted with AI's founders, and attended one of Newell and Simon's 1962 RAND summer seminars. Yet his vision of what computing ought to be concerned engineered artifacts, and remained very much outside the logical positivism and cognitive science which was the point of intellectual departure for most AI research at the time.

Engelbart's vision of the augmentative computer environment circa 1950, prior to entering graduate school, embodies most of the features of a development or computing environment available to programmers in the 1980s, or to any regular user in the 1990s:

" I had the image of sitting at a big CRT screen with all kinds of symbols, new and different symbols, not restricted to our old ones. The computer could be manipulating, and you could be operating all kinds of things to drive the computer. The engineering was easy to do; you could harness any kind of a lever or knob, or buttons, or switches, you wanted to, and the computer could sense them, and do something with it."

Well, I knew about screens, and how you could use the electronics to shape symbols from any kind of information you had. If there was information that could otherwise go to a card punch or a computer printer, that they had in those days, you could convert that to any kind of symbology you wanted on the screen. That just all came from the radar training, and the engineering I'd had, too, knowing about transistors. It's so easy for the computer to pick up signals, because in the radar stuff, you'd have knobs to turn that would crank tracers around and all. So the radar training was very critical, about being able to unfold that picture that rapidly.

... Just to complete the vision. I also really got a clear picture that one's colleagues could be sitting in other rooms with similar work stations, tied to the same computer complex, and could be sharing and working and collaborating very closely. And also the assumption that there'd be a lot of new skills, new ways of thinking that would evolve." (Engelbart interviews, op. cit.).

Engelbart foresaw these possibilities at a time when there were hardly any computers, even for the military or technical branches of governmental agencies, let alone ones which were user-friendly. Even a good decade later, when he began to develop the NLS (Online System) at SRI, he was routinely told that "what you ought

to research is how you can make computer work at all, these days" (RES emphasis). However, one caveat should suggest to us that it is not so very odd that Engelbart had a glimpse of a very different world of computing. As a trained technician and later an E.E. Ph.D., Engelbart had access to radar facilities. This technology was sufficiently important that it was relatively robust and lavishly developed, in contrast to computing's habitual progress in fits and starts. Perhaps because Engelbart, like Newell and Simon witnessing the SAGE program, saw the facile graphical depiction of data on a screen, the possibilities of computing seemed much more tangible. Still, we should keep the astounding modernity and freshness of his vision in mind.

Once he glimpsed the idea of a system that would fill the need for more information, faster, Engelbart tried to pursue it. He saw that he needed to study computers, and received a Ph.D. in EE at Berkeley as mentioned. He started a startup, long before it was fashionable, and then watched it fail, also long before it was fashionable. Stanford told him that computers were not important enough to be worthy of academic study, so he wrangled a post at SRI and worked there more than fifteen years, although he was at odds with the management during much of it.

Engelbart was subsidized from the early 1960s by various agencies, mostly military, who checked their suspicions sufficiently to place a small bet on a dark horse. Over the years, this roster included the Electronic Systems Division of the Air Force Office of Scientific Research (AFOSR), which would "give out little bits of money to all these different wild-haired guys"; the ROME Air Development Center (RADC), another long-time supporter of AI; and Larry Roberts, when the latter was a psychologist at NASA Langley Research Center and later in his capacity as IPTO manager (397). In 1963, Engelbart's premonitions anticipated those of the sometimes incredulous J.C.R. Licklider, another early mentor, as Engelbart began to build a working computing environment. He started by basing the environment on a mainframe setting. This was, of course, all that one could do before the integrated circuit had appeared on the commercial scene and gradually shrunk to a gloriously diminutive size.

J.C.R. Licklider had persuaded the Systems Development Corporation, the Santa Monica government computer company, to construct a large mainframe timeshared computer- “a huge machine, a many-million-dollar machine. It was just acres of bays of vacuum tubes and stuff; very fast and very complex and sophisticated instruction set for the time”. Licklider required other ARPA contractees to actually use it for their projects. Remote computing through modems had existed as far back as implementation at Bell Labs in 1940, and the Logic Theorist had been constructed w the help of a modem, so it was theoretically possible to work remotely. It was very rare, though: most of the contractees worked on the site.

For Engelbart, this was quite inconvenient, as it took him away from both his workplace and his young twin daughters. He and the other members of the Augmentation Research Project, as it was called from an early date, began using a modem to begin developing their text editing program, the core of Augment, from a more reasonable distance. The team added text editing, offline, using their own small Control Data Corporation 160-A mini computer. (This machine was intended to be the ‘buddy’ of the mainframe, handling input, output and peripherals without calling on the larger facilities of the presumably non-timeshared mainframe). The paper tape that served as IP and OP was displayed visually on a CRT display, edited on the screen, and then punched out again automatically. Paper tape was a fragile medium, but before the availability of magnetic tape drives or the pie-sized diskettes, which were used through much of the late 1960s and 1970s, it worked. The team even developed word wrap, in which the line of text is broken in two, if necessary breaking the word itself into more than one unit. This advance was sufficiently novel that Licklider was astonished at the demonstration:

“ Licklider came to see us another time and we showed him this and he would not believe it. He said, ‘Oh, it's not smart enough to do word wrap.’ He wouldn't believe that we had done it intentionally. He would just not believe it...” (Engelbart Stanford Interview 2).

By 1968 Engelbart and his team had created the core elements of the Augment system, also called NLS for **ON**line **S**ystem. All are

familiar to any computer user. Actually, they were more than familiar. Familial or ancestral are better words, since most of them have been surpassed. These features were: a keyboard; with a character-based display screen on which one may edit; timeshared usage of text files; pull-down menus; icons for data and application programs; the usage of macro-operators; and email including transmission of text and graphic files (the Journal feature of the NLS, developed in 1970 or so). One thing which ought not to be taken for granted, but that invariably is, is the computer mouse. The first mouse was so much bigger than a current mouse that one is tempted to call it a rat. The ARC worked on various versions before finally coming to one that closely approximates in size and movements those in usage now (398).

Several features of the Augment-NLS environment were highly subtle and not widely known. One was its extensibility. This aspect was present at the very start of Engelbart's vision, in which people would be able to define their own functions in accordance with their needs in terms of organizing information. It appeared as 'Verb/ Noun' in the NLS system (399). This would appear to be ancestral to macro operator commands, which once were a difficult and arcane feature of programming languages, now can be created by every customer service technician on Microsoft Word's helpline. Another special feature was the remote development of much of the system itself. A newer computer was available at the University of Utah but not at SRI-ARC yet, so the ARC team would log onto that one, running on the ARPAnet.

Engelbart and the ARC team established and ran one of the earliest forms of email and an electronically accessible library, as well as setting up and operating the first Network Information Center. Journal, the email system, shuttled messages, graphics and text to individual recipients and to designated groups. The Journal system also differed from most email in that it had permanence. The mail was also a library, a permanent record, rather than being intentionally erasable and usually erased rather fast, or held onto awkwardly, by the recipient. This was a first; Engelbart reports that it was not until 1972 or 1973, several years later, that BBN introduced somewhat comparable message systems. The email system organization was very simple, consisting of registering a

score or so of people on the ARC's computer address, and 'moving' them and rerouting their messages when they worked on a different computer. This was not too difficult for a group of fifteen. It was done, in fact, by one human being, a computer service operator working as a social secretary. But it did not provide a facile way for individuals to work at different computers should their steady habitat crash for the weekend, and it would be unworkable or at least computationally expensive in a computer context in which people were assigned computer facilities by queue, for instance.

And this was the Achilles heel of the original NIC and the email system too; neither of them was intended for large- indeed, ultimately enormous- numbers of participants. Neither of them scaled up. In 1972 or 1973, apparently, Larry Roberts and others began to push harder for the inclusion of more military nodes on the ARPAnet, and this quickly led to a need for a bigger email and address system. That came later, with the familiar DNS (domain name system), which proved sturdy for a long time as a means to designate entities on the network, was intended for and was successful with these far larger audiences. However, DNS itself was designed by former ARC member John Postel, after he moved to PARC. There is a strong case that the entire system was thus set in motion by the ARC (400). Moreover, it is not clear that networking and shared library resources were entirely popular in the mid- and late 1960s, when the ARC began to develop them. Roberts, Licklider and some of the BBN officials involved frequently saw their effort as unilateral. Without delving further into the politics of the issue, it would certainly seem again that Engelbart and company were far ahead of their time.

Like many of the ideas which Engelbart and the ARC group came up with, the idea of a shared library of computational resources may seem rather obvious to the present-day readers. But it was not at the time:

"...The ARPA office would get together its principal investigators at least once a year and sometimes more often. They would all have a "show and tell" session that would be quite fun. There were eight of us, and then ten and then twelve. By '67 there were 13 or so. We met in the spring of '67 at the University of Michigan in Ann Arbor with Bob Taylor, who was the director of that IPT office, and Larry Roberts, who had come down some months before from Cambridge

to be his deputy. Larry came with the idea of networks; he had been experimenting with them at Lincoln Labs. They said they were going to start a research project on networks. So everybody listened and said, "That's all very nice." But then they got to a point where they were saying, "We're going to connect all you people together with a network." Now that was something different. Everybody started sitting up in their seats saying, "Well, damn, I'm doing this very important research in artificial intelligence or in time sharing systems or something. I don't want to fool around and waste time getting all involved and getting my people involved with networks."

Adams: They weren't making the connection that if people got involved in networks it could facilitate their own work.

ENGELBART: Not at that point. They just got alarmed about it colliding with their interests. It was a very interesting dialogue that went on. They were being told, "Look, you can share resources." All these new kinds of concepts were coming in. It takes everybody a while to adjust. Bob Taylor happened to mention networks to me some months before, I guess the summer before. I was thinking about all that and said, "Why would anyone want to do that?" I remember saying that. (laughs) About an hour later I was thinking, "Gosh, what a funny reaction on my part." Because with a little reflection and a talk with him, I realized what it could do and how it would fit into the community goals I'd been thinking of. So anyway, I was much more prepared when it happened. I didn't realize they were actually going to start a program, so I was as surprised as anyone else when they announced it. I can paraphrase some of the reactions among the principal investigators. "You can share resources," says the ARPA office. So investigator A turns to B and says, "What have you got that we could ever use?" And this is very insulting to B, because of course his research is so important to him. Investigator B turns to A--they're all quick-witted guys--and he says, "Well, don't you read my reports?" This gets A because, of course he doesn't read the other guy's reports! And he comes back very quickly, "Do you send them to me?" And this gets B thinking, because he doesn't know where they go. Then they both realize they have a common problem. They both turn to the ARPA office and say, "You've got to set up a library of all our reports and all the resources, so we have some place to turn to know what's available." This stops Bob Taylor and Larry cold, because the two of them and the one secretary are already way overworked; how were they going to do it? So for a while that flounders around and they slough off that topic and talk about the different technical opportunities. I sit there and think, "Damn, that's a marvelous opportunity. If I volunteered to form the library, there's a community. But if I go back home and tell my people that I have committed us to that, it would be a problem. We worked things out by consensus." But it just got more and more intriguing. **Finally I volunteered, "Well, I'm interested in it. How about if I form an on-line library (I don't know if they called it the information center at the time) and run it for this community?"** It ended up that my research was not that big a distraction. It interested me anyway and everybody was relieved I was doing it.

Then it slowly got the ARPA office interested in this as something that was relevant to their own pursuit. It was three years before things were really operating, so in the interim I did a lot of thinking and planning. By the time things started in 1970, we had our Journal up, and the mail system, which for me was a big part of how you support the people in the system." (401)

Many features of the Augment /NLS systems seem to bear the visage of an instant classic, which would forever seem to fill, but not to over-embellish, all of the requirements of its genre- a sort of Bauhaus house or first Volkswagen car. Or so it may have seemed to those who saw it early on. Engelbart presented his work to the small and generally approving audiences of the early ARPA Principal Investigators' meetings, in the mid-1960s. He believed that the AI people were generally not very interested in the development of better computer environments. This impression does not appear entirely accurate in the long term, given the 'hacks', utilities for better text editing, display screen improvements, and other 'non-intelligent' changes to computing that came out of the big AI labs (especially MIT and Stanford). Be that as it may, Engelbart found that he had to wait a bit for a larger audience.

However, he got one rather abruptly when he finally presented NLS at the 1968 AFIP (American Federation of Information Processing) conference in San Francisco. The 1968 event was one of Engelbart's rare and apparently anxious efforts at better public relations for the work. It required elaborate arrangements, since remote networked displays of interactive computing involving timeshared online editing were rare at the time. The display took months of preparation and an hour and a half to proceed, but it did put him on the map- rather than only in SRI's doghouse. Engelbart's presentation of his system appears to be perceived as a genuine historical event in HCI: "There are more people who have claimed to have been there than were!" He told of 'people [who] came rushing up onto the stage' afterward. Others, such as Alan Kay, saw the session as a tremendous inspiration to proceed with interactive computing, if not to follow in Engelbart's path.

The event was the computational equivalent of the Chinese Revolution's Long March or the last helicopter out of Saigon in 1975, or the Columbia University riots. Almost no one who could

possibly have been party to the event will deny that they were present. It's impossible to check alibis.

Douglas Engelbart thus invented much of the software environment found in a personal computer system today. He did not come up with any of the hardware features, or any distinct computer terminal at all, and he never deviated from a vision of a small community of users with a common text editing and email system. This may have been his undoing, because he gave up opportunities to develop the Alto, a project which apparently caught fire among so many people.

SRI had apparently not been a perfect fit with Engelbart for some time. Dickson mentions that the ARC people had snaky Zapata mustaches, and some had long hair, but more than this must have been at issue. Engelbart's troops began defecting at about the time that PARC opened its doors in 1970. Larry Roberts had been their biggest fan when he was at NASA Langley in the early 1960s, and had helped them further in his capacity of IPTO director at ARPA. But help from ARPA dissipated for personal and professional reasons; ARPA wanted him to bring his work to a broader audience, and to integrate technically and possibly him personally, into a broader audience. When Roberts finally arrived at PARC, he brought in at least fifteen of Engelbart's team members. This was PARC's gain, if not Engelbart's own, as he stayed at SRI until the ARC unit was sold to Tymshare and the project practically fizzled.

Talent will out, however; Engelbart has received universal acclaim, including the 2000 National Medal of Technology awarded by President Clinton (402). He had believed that he would never fit into a university: an untested hypothesis. But he also never fit into SRI. He was the furthest thing possible from being a company man, and this benefitted his ultimate contribution, even as it derailed his career at SRI. The accolades that he continues to receive refute his eternal lack of rootedness in any academic or research institution.

Part IV. The Challenges of 1970

"May you live in interesting times". Medieval Chinese proverb, intended as a curse.

1. Introduction

The conclusion of the 1960s and the start of the 1970s were interesting times indeed, for their general qualities of political upheaval, intellectual radicalism, and thus ultimately less certainty for research funding.

Through the agency of the latter, technological developments were directly affected by the winds of political change, as we will see in **Chapter 8** of this segment. Political volatility helped lead to a far more restrictive- we might say obtrusive- research environment for AI. This in turn was consequential for actual research results. It would force the field- knowledge engineering in particular- to produce practicable real-world applications. These will be discussed at length in Chapter Nine. Moreover, cultural upheavals of the 1960s were not relegated to pop culture (**Chapters 9 and 10**).

2. Rage Against the Machine

High culture in the United States spent the decade of 1965 through 1975 consumed by a peculiarly extreme self-loathing. One common critique of science and scientists during the larger Postwar period was that they were estranged from high culture. In this case, it is just as well. A hodgepodge of dissonant artwork, clashing atonal music, and the insistence that the suffering it entailed made it real and meaningful- made for a strange cultural *mélange*. The specter of nuclear warfare was indeed a real one. An enormous number of people who grew up in the Fifties and early 1960s (the prime ages of today's baby boomers, or the 'postwar demographic bulge') were truly terrified of the Bomb. This 'knowledge beyond forgiveness' is very difficult for anyone who did not grow up in its shadow to grasp. This brought an intensity to the debate that is hard to capture in today's very different political environment.

During the late 1960s, the societies of all industrialized nations and certainly that of the United States, were shaken by a cultural cataclysm as heady as anything since the second decade of the Twentieth century itself. These years witnessed the falling away of any putative belief in cultural norms on the part of the general public. The questioning- sometimes a spurious questioning- of such

norms, was accorded much respect. The objective facts indicate that science and technology marshaled on and that airplanes continued to fly, and scientists continued to study oncology and mechanical sciences. Despite this, the norms of the cultural upheaval declared that progress was being simply overshadowed by a spiritual awakening deeper than any scientific understanding could fathom. The latter proposal in particular is pernicious in that it is not falsifiable: if one does not agree with it, then the putative lack of spiritual understanding is clearly at fault. This possibility was given more than its fifteen minutes of due attention.

Science and technology largely ignored this cultural vogue, instead manifesting numerous achievements from the respectable to the revolutionary. These include the Apollo Moon launch, subsuming numerous aerospace achievements in the process; the first heart transplant. Among these technologies was AI, which actively charged ahead on a blue streak of achievement.

3. The War at Home and AI

The military has been central to our story thus far, public opinions as to its activities have not been. However, even AI did not escape the cataclysm that shook the campus and civil society in the United States throughout the end of the 1960s and the start of the 1970s. The United States Army's efforts over decades to defeat the North Vietnamese Army (the Vietcong) was a tragic black hole of human lives and materiel, which escalated rather than ended during the 1960s. Anti-war sentiment ascended to a fever pitch, and the pot boiled away at an intolerable whistle for years. Disapproval of the war spiraled into disapproval of the "military-industrial complex", capitalism [sic]- and many other targets: the nuclear family, the Protestant work ethic, navy blue, proper grammar, short hair, decent hygiene, being on time for appointments, dentists, etc.

Eventually, the list of grievances against "the System" became almost too broad to characterize at all. American universities, as putative props in the "military-industrial complex", were subject to ceaseless protests. The prime focus, if there was one at all, was ending the war, something that university administrators had no power to do. Subsidiary demands were usually the breaking of any

of several key links between universities and the military- typically ROTC officer training facilities (403) and classified or weapons research on campus. The apparent goals of some of the more intellectual of the student protest groups, such as Students for a Democratic Society, included the demilitarization of the campus. But reaching toward what they saw as the root of the problem was hopeless, for the whole PostWar edifice of science and technology was entwined with the Cold War.

Still, the sweeping nature of protests disrupted higher education during 1969 and 1970. While MIT was somewhat inconvenienced at times, those students most closely entwined with the military-industrial complex were not the most likely to become enraged (404). Instead, intensity of protest appears to have been a function of the presence of intellectually and economically elite humanities students, and of the size of the undergraduate population. Stanford and CMU were somewhat affected, while Berkeley, Cornell, and Columbia were almost closed down entirely by repeated, large-scale protests. At Kent State and the University of Wisconsin, the student protests led to the deaths of otherwise uninvolved people (405). Fortunately for AI, mass protests were never intellectual enough to be focused upon AI, or even computer science in general: “The Machine” as an abstraction was vilified, but evidently “the Machine” as a concrete entity was spared.

Stanford University

According to the informal formula suggested above, Stanford was a natural target for the unrest of the late 1960s. Stanford had been established as a liberal arts institution during the nineteenth century, and maintained its sense of elite engagement with esoteric humanities field. This foundation was only overwhelmed in importance by the addition of the Engineering School (in 1921) and the growth of the engineering fields after the Second World War. Moreover, at this time Stanford University formally included the Stanford Research Institute. As we saw in Chapter Seven, the Stanford Research Institute had been established as an academic think tank to aid “Pacific” businesses. This noble aspiration had been swamped by a ‘take-all-customers’ managerial philosophy that naturally embraced defense contracting. In the mid-1960s, four-

fifths of SRI's revenues came from the government; and of this, more than three-quarters came from the military (406). These contracts included research on chemical weapons and counter-insurgency tactics, and the university's R&D docket included instruments of destruction. This proved intolerable to the increasingly pacifistic or at least anti-U.S. political agenda. Stanford students, angry at the discovery that the Stanford Electronics Laboratory had helped the CIA with surveillance projects, began picketing Stanford's administrative offices, as early as May 1966. Soon, the Students for a Democratic Society were holding protests at SRI, a mile or two north of Stanford's campus. The actions escalated, and the SDS and sympathizers seized Stanford's Applied Electronics Laboratory for nine days, using the building as a child-care center and for sundry other civilian purposes (407). Bruce Franklin, an angry- and tenured- English professor, tried to incite a "people's war" against the Stanford Computation Center- and was fired from Stanford for this.

Protests continued, including physical attacks on then-Stanford President Kenneth Pitzer, and demonstrations that led to physical injuries. It did not stop there, and the university administration responded by making changes. The Stanford University trustees formally divested the university of the Stanford Research Institute in 1969. The Institute was sold to its Board of Directors for twenty-five million dollars, and changed its name to SRI International (408). The ROTC program on campus was stripped of academic credit-granting status. Finally, the university announced that it would no longer accept grants or contracts for Classified research (409). Ultimately, protests died down, but not before Pitzer resigned, having appeased neither liberals nor hawks.

John McCarthy seems to have been also affected by the 'Time of Troubles'. He had been a red-diaper baby, but had broken with the Old Left in 1953. However, his rejection of Stalinism did not preclude sympathy with the New Left at Stanford. He procured five thousand dollars for the Mid-Penninsula Free University. At the first meeting a group of 'ultra-radicals' showed up and forcefully dominated the meeting. McCarthy visited the Soviet Union and Czechoslovakia just before the 1968 Soviet invasion. Both experiences turned him into an anti-Stalinist, suspicious of radical

leftism (410). He went further in the early 1970s, cutting his hair short and wearing suits, provoking radical chic with his contrarianism: “I didn’t become an actual reactionary until about 1972...” (411).

MIT

MIT’s identity with regard to military spending and technical research was not at all ambiguous. It was both a defense research center and a university, as we considered in Chapter Five. The potential grievances were much more extensive, but the actual outbursts were far more mild. However, forty-eight MIT faculty members did take the astounding initiative of calling a strike in January, 1969. The only one associated with AI even tangentially was Noam Chomsky. Given his political activism, it is not surprising that he was concerned with MIT’s identity in a decade of endless war. Chomsky stated, as quoted later in MIT’s Review Panel on the Special Laboratories,

“In an institution largely devoted to science and technology, we do not enjoy the luxury of refusing to take a stand on the essentially political question of how science and technology will be put to use, and we have a responsibility to take our stand with consideration and care.” (412)

In April, 1969, a number of agitated students confronted Instrumentation Laboratory founder Charles Stark Draper and picketed the Laboratory, and created adequate adverse publicity that the Laboratory was renamed after its founder and was amputated from MIT in 1970 (413). Perhaps because it was twelve miles outside of Cambridge, Lincoln Laboratories, remained part of MIT and federally funded. It was rumored that one of the antiwar protests would lead to the very doors of the MIT AI Laboratory. In anticipation, the doors were fitted with locks for the first time, glass panels were replaced with bulletproof Plexiglas, and hackers rigged the elevator so that it could not be taken directly to the laboratory. The security measures proved unnecessary- although none of them were reversed (414).

The Lesser “Troubles” at Carnegie-Mellon

Carnegie-Mellon's "Troubles", as Herbert Simon describes them, had no effect upon the structure of the university: CMU was a more civilian entity to begin with. This does not mean that it did not receive its share of student discontent, but that the issues in the skirmish were more generic to U.S. society. The height of student radical activity were non-violent skirmishes over the University's lack of support for the desegregation of construction jobs for the building of a new science center. The issue was resolved in favor of a racially mixed construction workers' union. CMU narrowly averted an attack on its ROTC building following the killings of students at Kent State University by National Guard troops in May, 1970 (415).

Around this time, the student radicals began to irritate Simon, with their 'incivility' and their calls for "world revolution and the seizure of power from those of us who were over thirty" (416). Simon especially objected to the campus Maoists, "mainly in the Architecture Department". He proceeded to incite the latter faction. Arguing that "a kept press cannot be a free press", he called for an end to the University's subsidy of the self-proclaimed 'Maoist-run' campus newspaper. Trying to drown out their writings on 'revolution', he wrote his own column for a number of weeks (417). The student newspaper editors' bark proved louder than their bite. Simon literally out-wrote them. They asked Simon to stop publishing "Simon Says" before he had even presented his program of university reform (418).

The protests against the Vietnam War were perhaps the only 'successful' element in this wave of popular insurgence. This was a perfect example of a Pyrrhic victory, in which the victory's price is too high for any satisfaction. The protests were successful in achieving their putative goal, not successful in that the denouement of the Vietnam War was terrifying. In 1974, the United States direct involvement in Indochina ended: the U.S.A. withdrew, and the North Vietnamese Vietcong military regime took over the whole of the newly united country. The Vietcong subsequently proceeded to prove true every prediction made by "U.S. imperialists" by forcing educated people into labor camps and forbidding rudimentary private enterprise. This is one case, however, in which "I told you so" offers no enjoyment. The student protests were effective in this

particular case- although the historical perspective on this issue over even two decades is deeply distressing. The universities, and particularly the most distinguished ones, were closely entwined in the larger military economy. AI people were not generally weapons-builders, or particularly gung-ho on the American role in the Vietnam War. But the familial identity of AI made such a heritage inevitable. The extrication of AI from the military was not even a theoretical possibility.

Superficially, the troubles on American campuses and in American society in the late 1960s seem to have passed AI by. However, the upheavals of the 1960s did not originate or conclude with the anti-War movement. The term itself is a reference to a number of independent movements, one of which certainly was a rebellion against the extant Classicism in psychology and art. This in turn would eventually affect AI by introducing science and AI engineering which emphasized the non-cogitational features of intelligence. Graphical rather than text depiction of data, attempts at depicting human sensory capacities, and a less grand view of the human mind itself, are several manifestations of the Romantic or non-Classical view of the mind.

Chapter 8. The Mansfield Amendment, “The Heilmeier Era”, and the Crisis in Research Funding

The Crisis in AI Research Funding in the Early 1970s

“The mountains are high and the emperor is far away.” Medieval Chinese proverb.

Well, not any more.

The baker’s dozen of AI researchers had a happy childhood in the fully subsidized research world of the 1960s. Subsidies for productive scientists were easily arranged in the first decade of the Advanced Research Projects Information Projects Technology Office. Procurements for research in the 1960s in particular had been insouciant, in accordance with the Federal government’s dedication to space exploration and with the soaring military expenditures of

the Vietnam War. The AI community was so small that peer review was practically redundant. Instead, according to Minsky, a program manager could simply undertake an intuitive evaluation of the zeitgeist. Formal proposals were not needed; most monies were promptly dispatched following phone calls to the right person. As we have seen, ARPA and IPTO did indeed try to propel forward certain sorts of computing infrastructure, and the hand on the checkbook was remarkably light. Edward Feigenbaum refers to the research world at the time as "paradise".

Indeed, it is possible to write and read about the history of AI, including its governmental affiliations, without remembering that the United States was at war with North Vietnam, albeit unofficially, through 1974. AI had, in fact, been born directly from the arms race, in the visage of the SAGE project. But this lineage proved to be easy to push into the background. The intentions of the AI researchers addressed sufficiently basic questions at a sufficiently theoretical level that they sidestepped the arms race going on in a state of cold fury at the time.

It is almost possible to read about AI's development up to this point as if the scientists lived in an ivory tower, because they did. But if the scientists did live in an ivory tower, and should have, their DARPA program administrators did not. The testimony which ARPA Director R.L. Sproull, presented to Congress in 1964 proffered information processing as a significant technology for the Armed Services:

"... this research has implications of significance to other areas of defense department endeavor, including military planning, logistics, and research as well as the are of military operations." (419).

One obvious and immediate technological windfall from the ongoing justification of computing to the military was the Internet. Distributed computing was presented as a boon to the need for redundancy as a protection against geographically concentrated military installations; the ARPAnet was indeed presented as a means of secure communications in the event of a Soviet first strike. AI was offered as a technology which could help military information systems. Timesharing, superior computer graphics,

including bit-mapped screens and the display of photographic-type images and images which rotated, were presented by DARPA officials as technologies which could truly improve the logistics of military operations (420). As J.C.R. Licklider himself proposed in a 1964 book on the topic, AI was ideally suited to embed intelligence in command and control, and problem-solving programs in AI could work out much of the thinking for such MIS systems (421). While the ulterior motives for the arguments of Licklider, and perhaps others, may have been ultimately intellectual, it is clear that the easiest political role for the field to play was as part of the loyal effort to maintain the United States' vaunted primacy in matters military and strategic.

The Defense Community in general, and certainly ARPA in particular, presented themselves with a considerable sense of omniscience and oversight over the entire 'Free World' during the middle of the 1960s. In 1965, the House Subcommittee of the Committee on Appropriations, treated the Director of DR&E, Harold Brown, as if he was Einstein. Another famous DoD figure, Robert McNamara presented a grand overview, as magisterial as that of Queen Victoria or any Czar, of the United States' management of the Cold War challenges in each and every corner of the planet. The overview is so comprehensive that it includes Yemen, decades before Al Qaeda. It is not surprising that both- Harold Brown and Robert McNamara, that is- were given practically everything they wished for, money-wise. This largesse was apt to run out at some point, and indeed it did. But much of that changed, over several years, starting late in the 1960s.

In retrospect and even at the time, it is hard to say that the Cold War and the arms race against the Soviet Union were wrong, in principle. But principle was broken on the wheel of practice. Increasingly in popular opinion at the time, one could not deduce from this premise the conclusion that the Vietnam War, especially as it was being fought, was right and just. A gradual and widespread souring of popular sentiment concerning military spending took place in the late 1960s. By the end of that decade, every form of military activity, including relatively pacifistic research such as the AI research carried out by ARPA, was under the looking glass. Being leaned on, the military had, among other things, to write smaller

checks to ARPA for a time. The military, in essence, finally wished to cash the blank checks that it had written to AI all these years. This was not the military's idea, so to speak, but was a result of intense pressure and political angst in the United States per se. The existence of 'war at home' as the student unrest of the turn of the 1970s has been called, is well-established. Some of the sentiment behind popular unrest and campus demonstrations was expressed in a more genteel form in contentious Congressional hearings.

Anti-war and anti-military sentiment were fueled by other things such as a vague pop culture version of phenomenology. Merged with fear of nuclear warfare, awe at the landing of men on the Moon, and with the increasingly pervasive opposition to United States involvement in support of the South Vietnamese army, the cauldron was a near-apocalyptic political crucible.

The Vietnam Conflict was not pretty. As the song lyrics from the Broadway musical Hair put it, it was indeed "a dirty little war", tragic and fraught with cruelty from any political perspective. ARPA was, unfortunately, more part of it than current visions of the Agency would wish to acknowledge. ARPA's expenditures, historically, have gone to such things as defoliants, better methods of detecting underground nuclear tests, and designing smaller helicopters and lightweight body armor. ARPA even maintained its own laboratories in Saigon and Bangkok, for the counter-insurgency study called Project AGILE, during the better part of the Vietnam War (DOD 1965 hearings) (422).

AI per se was involved insofar as MIS systems were involved, although this is a research issue that the author can only mention in theory. But even if AI was understood by all of its research scientists to be a scientific topic, the topic's massive governmental support had to be justified as a potential aid to military endeavors. Therefore, when all research under the military auspices was placed under scrutiny, AI was too.

The Buck Stopped Here: the Mansfield Amendment

Continued funding of military research at major universities was itself placed in jeopardy at the end of the 1960s, as Congressmen

questioned whether they should grant money to MIT, for instance, when even that institution was subject to serious student unrest (U.S. Senate Subcommittee hearings, 1969) (423). The latter sniping from the Congress continued for several years. This could be countered fairly easily, and it was, by the inimitable longtime DDR&E Director John Foster. Facing increased skepticism, Foster and the other ARPA officials (Lukasik, Rechten, and Russell Beard), became notably more pointed, defensive and strategic in their justification of investment in long-term research in computing and electronics. (Arpa changed its name to Darpa- 'Defense' Advanced Research Projects Agency- in 1972, according to the agency website). The value of such research has been paid back many times over, but this was not a foregone conclusion in the late 1960s and arguments as to the long-term military and economic value of computing had to be made repeatedly. Congressional hearings following 1970 typically included a series of pointed questions as to the target end dates for any given research. Moreover, the ARPA administration shifted research projects into the more 'pragmatic' designation of "exploratory development", possibly in hopes of seeming less impractical.

The most immediate threat to the continuation of things as they had been was the Mansfield Amendment, a restrictive rider inserted into military authorization bill PL 91-121 in 1970 (424). The Amendment was an attempt to make Defense Department research more accountable to Congress, and it did achieve this end, to some degree. Senator Michael Mansfield (Democrat, Montana) engaged in heated arguments with John Foster, in Congressional hearings in 1968 (425). Foster held his own, insisting that civilian research, under the bounteous and elastic auspices of the military agencies, would result in significant military advances. Moreover, Foster, insisted, as would his counterparts for the next two decades, that at any moment the Soviet Union was going to catch up with the United States in military power, and that the military and technology spending was necessary. But Mansfield was not impressed, and the next year he added a rider to the appropriations bill which paid for the DOD's expenses.

Compliance with this legislation was harder than it sounded. First of all, the wording itself had to be figured out more precisely. In

addition to this, Mansfield, and those who supported him and voted with him, clearly were interested in pestering the Military establishment in more ways than one. Mansfield himself proposed that scientific research be spent through the National Science Foundation rather than the DoD. This was an impractical idea which had been vetoed even by former general such as Eisenhower. Mansfield instigated GAO scrutiny of the ARPA projects under the DoD auspices, requested that the OST help assure that the projects which did not belong under the DoD umbrella be moved to the NSF area, and otherwise ascertained that projects would be more carefully scrutinized (426). Five years later, the GAO was still analyzing DARPA's internal accounting system and finding minor faults, and Congressmen on the Committees on Appropriations were grilling DARPA officials, such as Stephen Lukasik, on this issue (427). From the point of view of AI strictly speaking, Mansfield's sardonic reminder that "All that is required under Section 203 is relevance, which is not a dirty word as some critics of the section sometimes seem to suggest..." (428), meant the beginning of a more irksome and strenuous participation in the political process of scientific grantsmanship.

The Mansfield Amendment itself required that the wording of DoD and DoD-research-related documents become much more precise and more explicitly devoted to expressly military ends. Long-term research would hence be tethered to a much tighter short-term horizon. For the first time, in some cases, this meant that AI research previously paid for by DARPA had to be thought through for 'relevance', even tenuous relevance, to military ends. The AI scientists, no longer insulated from wider political circumstances, were now put under pressure to produce 'useful' technical devices, or at least to speculate as to military applications for their projects. Probably only coincidentally, AI in Great Britain faced similar problems around 1970. At about the same time as the Mansfield Amendment was passed, the British Government issued the Lighthill Report, a white paper which contained scathing, if poorly-substantiated, proclamations as to the impossibility of AI. The political fallout resulted in curtailed monies for AI research in England.

John McCarthy observed that the terms of the administration of AI did indeed change: "About 1970 they got increasingly short range- 'What are you going to do for us now ? In the next two years?' Still, this was not particularly substantive for established participants: McCarthy observes that the rules changed in the early 1970s, but the actors didn't change:

" If you were to look at the successive doctrines, what you would discover is that the actual work that they supported fluctuated much less than the doctrines fluctuated. The people who were supported were on the whole a pretty stable group, much more stable than the people who were supporting them, for example." (429).

The actors, so to speak, were evidently irked at having to audition for parts that they knew they would get to play. Minsky and Papert found the submission of detailed proposals so irksome that they resigned the administrative post of directing the MIT AI Laboratory in 1972 (430). Their successor, the young Patrick Winston, admits that the harsher policy made him a more sophisticated professional scientist. He notes that the new laws and the new IPTO leader required that people who wanted grants state prospective research milestones and points of relevance to DARPA objectives. The amorphous phrase, "continue to work on" was replaced by wording now commonly used: "report on," "test," "bring up to a measurable level of capability," "intermediate measureables". Prospective commercial applications could be conceived with careful consideration:

" It wasn't enough to study reasoning; you had to talk about how it might be applied to ship maintenance or something." (431).

This was not as difficult as it sounded, and to anyone who has ever applied for a highly competitive grant or position, such grantsmanship would simply have to be seen as part of the larger research project. Indeed, figuring out how to present one's proposed work to the client is part of the cost of doing business for most people, which makes the complaints of AI practitioners seem a bit precious. Ultimately, the 'hardship' of the changes in terms of research propelled the field to move forward toward commercial

applications. At the time it was treated as cruel and unusual punishment.

Genuine Threats During the End of the Vietnam War

Despite the complaints, this was no time for irritation. AI was in more trouble than most of its practitioners were entirely aware of. The continuation of federal research funding for AI itself was being reconsidered. The ugly and protracted ending of the Vietnam war had ushered in an era of profoundly negative attitudes toward the government in general and military spending in particular. This in turn compelled military research leaders to regroup their forces for the political 'war at home', so to speak, and to rather quickly proclaim the need for a new and more technically sophisticated Armed Forces to counter the ever-growing Soviet military threat. The defense research administrators to undertake a knotty and difficult redesign of the rationales for military research. In the process they faced a weary and jaded Congress on one side, and an impatient and possibly disrespectful (or so they said) group of AI research leaders on the other.

The twin problems of acting as liaison to both of the above parties helps to explain why even finding program managers for DARPA was difficult during this decade. Another problem was internal to the AI community- or perhaps to the community of scientists per se. Government service became less attractive to many people in the university community during the 1970s. Some perceived DARPA-type positions as being paid very little money in exchange for fighting with Congress, the military and bureaucrats. Irritation at the meagerness of its financial rewards in a decade of increasing costs of living was only one reason for the anemic supply of warm bodies with doctorates and the right contacts for IPTO and DARPA jobs. Others simply took advantage of opportunities in the giddy new commercial computing markets (432). Some, such as Robert Engelmores and much later Edward Feigenbaum, simply did not commit themselves to staying very long in civil service.

Thus, of the handful of top people who were qualified to take these jobs, most did not wish to. When IPTO Director Larry Roberts prepared to leave IPTO in 1972, no one was willing to take his place.

J.C.R. Licklider had to be recruited back to IPTO, where he served as Director again from 1974 to 1976 (433). When Licklider left for the second and final time, Heilmeier replaced him with longtime ARPA program manager, and former Pentagon colleague and nuclear physicist, David Russell (Heilmeier CBI Interview OH22, 1991; also Congressional hearings) (434). Many of AI's practitioners apparently would have preferred that decade's Latin American generals as their bosses to the American ones they were given. Russell stayed three years, and in 1979, consummate insider Robert Kahn took over IPTO (435).

George Heilmeier, who was director of DARPA from early 1975 through 1979, was initially wary of the contributions of AI (436), and took a number of steps that could only indicate this. However much it disturbed AI's scientists, it is not surprising that anyone enduring intense pressure from above, as Heilmeier was, would not be suspicious of technologies which were labeled "exotic" and were apparently not always useful. Heilmeier's background was military-industrial in general, but not scientific or related to AI in specific. He had joined the Pentagon in 1970 as a White House Fellow, and then moved into the office of the Director of Defense Research and Engineering (DDR&E), eventually as the assistant director (437).

Heilmeier exerted much care about AI's proposed activities and real achievements, both because of clear personal inclination and perhaps because of the mandate from above. He read and signed DARPA orders himself, in the process questioning their merits. He asserts that other DARPA directors had not done this. Worse, he determined that AI in particular needed much closer scrutiny brought to bear upon it:

"They were getting a large amount of money, and for the life of me I couldn't tell what they were going to do."

Moreover, when Heilmeier insisted that they tell him what they did in more detail, he enraged the AI community. He was told, he says, that;

"You are going to destroy the community"- and, he asserts, "They said that I didn't have the right to ask the questions because I didn't

really understand their field. In essence, 'You aren't smart enough.'" (438)

The Jason Committee Inquiry

Heilmeier responded to criticism of his own intelligence by people inside AI, and to criticisms from outside of the possibly egregious overspending on AI, by subjecting AI to the scrutiny of a Jason committee during the summer of 1975. The Jason committee, an elite military advisory board in which the leaders of AI occasionally participated, gathered in the scenic resort town of La Jolla, California, north of San Diego (439). Certainly all of the Jasons were indeed smart enough. Not only were they smart enough, but they were a packed jury. A number of the top AI people- Minsky, Newell, McCarthy, Moses, Winston, Lederberg and Feigenbaum- were consulted during the course of the summer study. The Jason report, prepared in part by Saul Amarel, a Jason who was also historically involved with AI and was later DARPA director himself, apparently assuaged Heilmeier, but not enough to persuade him to go easy on AI.

The Jasons, in turn, were attached to the Institute for Defense Analysis, a Pentagon think tank which provided strategic analyses for the Vietnam and other conflicts. The top universities were even more integrated into the "military-industrial establishment" than one would think.

" A unique entity within IDA is its Jason division, composed of forty to forty-five outstanding university scientists and including several Nobel laureates. The Jasons, as they are called, are generally professors with tenure at places like Princeton, the U. of C., and MIT but spend much of their spare time, certain weekends and all of their summers thinking about warfare for the institute on a consulting basis. Each summer they and their families are whisked off to a remote resort where, under heavy security, they engage in intense group thinking on defense matters..." (440).

The Jason committee and Heilmeier's own inquiries found that AI was indeed a meritorious field, and nevertheless proceeded to elicit a very different style of grants administration from the AI

community. Heilmeier claims that he pushed the field's researchers because of his belief that " IPTO was [and is] sitting on the technology base that...could have a major impact on the Department of Defense. It hadn't had that impact" (441). Perhaps vitiating claims that he was indeed supportive of applied AI research, Heilmeier championed AI after he left DARPA and became the Vice-President for Research, Development, and Engineering at Texas Instruments. TI developed one of the first commercial LISP workstations, and introduced expert systems applications. Heilmeier asserts that his 'conversion' took place well before, not after, he arrived at TI (ibid.).

It is clear, regardless of the validity of the latter statement, that the relationship was strained beyond the point of civility. Marvin Minsky asserts that (unspecified) DARPA officials threatened the AI people with Congressional scrutiny if they did not meet the deadlines which they had set for themselves (442). It seems only reasonable to become irate at being driven from Paradise- there is historical, even Biblical precedent. But certainly we may observe an historical irony in this group's indignation, since their very privileges were a recently minted artifact.

This did not mean that Heilmeier blessed everything that had gone before. Indeed, some people regard him with as much warmth as a conquered nation regards the occupying army. A harsher judgement of his tenure would point to the complete cessation of funds to SRI's active and productive robotics programs. Nils Nilsson, head of the Shakey the Robot project at SRI discussed in Chapter Six, lost his funding. Nilsson was engaged in a long-term, highly speculative but very fruitful effort. For such things, "...DARPA was the only game in town on that one." When the DARPA money was halted, so was the project itself (443). Nilsson also lost his 'CBC', or 'Computer-Based Consulting', project, which was essentially an early expert systems foray, when ARPA insisted that it address the particulars of command and control more closely and the particulars of electro-mechanical equipment repair less closely. In Nilsson's own dry words, "I don't take redirecting too well". He resigned as head of that ARPA project (444).

Also at SRI, DARPA ended funding to the NLS project headed by Douglas Engelbart. NLS had been developing various prescient user-centered innovations, including a local area network, the computer mouse with a bit-mapped screen, an online editing program, and the rough equivalent of email, and all of these projects were abandoned. We should note that the innovations themselves were taken up at SRI (in the form of industrial rather than research robotics; Nilsson 1988), and at Xerox PARC (445). Finally, in 1975, DARPA ceased funding for ARPA-SUR, a variegated and productive project in speech understanding, with many products and an all-star cast of researchers. This move evoked considerable indignation in the community (446). This ire was exacerbated when Heilmeier left DARPA, joined Texas Instruments, and proceeded to rehire the now-jobless speech researchers (447).

The Redirection of Military Research and of DARPA

Heilmeier irritated and even infuriated many people. But with the benefit of historical hindsight, they appear to be brilliant leaders. They showed remarkable perspicacity and innovation in the ongoing bureaucratic battle for funding of research during the sump of American military machismo. During the mid-1970s, the ARPA administration in specific and the American military in general were engaged in a sustained and successful redesign of their own technological goals. The 'new' military discovered a considerable Soviet threat and a wealth of new sorts of weapons which could be developed to counter it. In 1977, Heilmeier told a Congressional committee:

“ During the past several years you have been alerted to the tremendous Soviet drive to seize the technological initiative as well as gain superiority in deployed military equipment. I don't like the trends, and I am somewhat overwhelmed by the sheer magnitude of the Soviet effort.” (448).

Moreover, the defense research establishment found remarkable new technologies and uses for them. In 1969, even Secretary of Defense Melvin B. Laird was obliged to apologize for the expenses of the DoD. He presented proposed budget cuts of one-half billion dollars for DoD with the conciliatory statement:

“ Thus the defense program in itself does not constitute an unreasonable burden on the economy.... As you know, President Nixon is doing everything he possibly can to bring that conflict [Vietnam] to an end.” (449).

But relatively soon after the Vietnam War actually ended, the military researchers were invoking ‘technology revitalization’ as a reason for increased funding to the defense agencies- not necessarily to the Armed Forces themselves- every man for himself here. ARPA had begun to propose, in an entrepreneurial rather than arrogant or self-assured manner, vigorous and rapid defense applications of computing and of new technologies.

Generally speaking, neither Licklider nor Roberts had ever been placed in the sort of economy of scarcity that emerged in the 1970s. Most of these were, of course, R&D fields leading to weapons based on physics or new materials or other fields. For instance, during this period the idea of ‘smart weapons’, which integrated AI functions and sensors, was introduced, as was that of high energy lasers in space and much more powerful detection of submarines (450). Moreover, the synoptic presentations of Heilmeier to Congress appear to be the first ones which suggest the control of military expenditures through widespread application of computing software.

This agile response to demand from the Pentagon, rather than acquiescence to the considerably more relaxed natural pace of academic research, is what infuriated so many people about Heilmeier and his colleagues. Instead of endorsing the agendas of individuals in the research community, Heilmeier attempted to bring elements of his own direction to the field. He did this in the form of difficult applications presented to the community. These ‘challenges’ included Morse Code recognition and the interpretation of sonar signals and ASW [anti-submarine warfare] signals. It also included Command and control- as Heilmeier stated in a Charles Babbage Institute interview, “ I wanted systems that could adapt to the commander instead of forcing the commander to adapt”:

“So I said, "Look, if some of you guys would sign up for these challenges I can justify more fundamental work in AI." And some did.” (451).

Heilmeier's 'challenge' hooks caught some fish- indeed, one could even say an entire school. The anti-submarine warfare problem was taken up by Edward Feigenbaum and HPP colleague, his former Dendral colleague and wife H. Penny Nii, who conducted classified research on the topic in 1974 and 1975. ASW, in which the data being given to the relevant personnel are highly heterogeneous, was also immediate and explicitly military. The application was needed for making sense of the miscellaneous data of Navy sensors in the Pacific Ocean, which were trying to interpret the movements of the Russian Navy (452). The integration of this data, fulfilled in the HASP and HARP programs in the mid-1970s, proved to be intellectually intriguing as well. The different sorts of information from sensors were properly represented in a heterarchical rather than necessarily hierarchical environment. Ultimately, Feigenbaum and Nii used the first version of a blackboard architecture, in which all inputs to a central 'blackboard' of data may affect the ultimate interpretation (453).

In addition to Feigenbaum and Nii's specific contributions to the U.S. Navy, and the more general contributions of blackboard architectures to AI, Heilmeier's "redirecting" initiative apparently instigated some more aggressive efforts within AI to develop fielded applications.

The Numbers on Research Funding

" A billion here, a billion there, and pretty soon you're talking real money". Attributed to Illinois Senator Everett Dirksen (1896-1969).

It is useful to think of ARPA's funds for AI as the smallest of a series of proverbial Russian dolls, nested in successive layers. The largest of these was mammoth: the expenditures for the whole DoD itself were invariably huge. Countless smaller sums were submerged in the amounts. The expense which is pertinent to us is, of course, defense research. This immediately cuts the magnitude of the sums down to a size which is more easily imagined. Defense RDT&E is budgeted within the defense agencies, a category that includes ARPA/ DARPA as well as the Institute for Defense Analysis, the Defense Information Agency, the National Security Agency, and the

Defense Communications Agency. The general title of RDT&E includes as well the research branches within each of the Armed Forces. We have already encountered several of these branches, for instance the ONR, the NRL, and the AFOSR. Despite the preponderance of attention paid to DARPA, each of the latter has historically funded AI. As we saw earlier, the RAND Corporation was originally set up as an auxiliary research agency for the Air Force, and there are other areas of Defense-related research (Lincoln Laboratories, the ISI) which are doubtless fundamentally contiguous to AI although not reflected in the official documents. The DoD-AI relationship is thus multifold. We shall not have to come up with any conclusive money figure concerning total federal support of AI research.

When we remove a further layer, the next level is much tinier. The defense agencies themselves, of which ARPA has historically been the greatest champion of AI, are only a small portion of the RDT&E performed by the military. Defense RDT&E is big money. Even in FY1963, the figure was roughly seven billion dollars (454). The entire defense agency appropriation itself often has totaled less than ten percent of all RDT&E within the larger DoD. In FY1978, \$779.3 million was allocated for all the defense agencies. The next fiscal year, the figure was much higher, \$954 million, but this still made up only 7.4 % of total RDTE expenditures. During the next two years, the Defense Agencies expenses kept rising, running at \$1120 million for FY1980, and \$1234.8 million in FY1981. But this expenditure was still well under 10% of total RDTE (455).

As we remove the cover of the larger Russian dolls of RDT&E and the defense agencies, another surprise awaits us. Within this category, DARPA itself is relatively small, although it obviously provides a good value for the money. The figure inside, that of allocations for AI, usually under the auspices of the IPTO, is far tinier still. DARPA has historically commanded a respectable quantity of money for research projects, which is a small amount of money for the Defense Department. Rather than remaining the same or rising with inflation, the amounts have also been subject to fairly drastic fluctuations in both directions. Typically, the amounts allocated have been relatively consistent with appropriations. At first, ARPA allocations were quite substantial, since space projects were placed

under its umbrella. At that time, the ARPA budget exceeded \$300 million, totaling \$520 million in FY 1959 and \$455 million in FY1960 (456). Once NASA was formed, the figure dropped to the neighborhood of 200 to 250 million for the first half of the 1960s.

DOLLARS APPROPRIATED BY FY; in millions;

1958	83 1
1959	520.0
1960	455 0
1961	215 0
1962	186 0
1963	250 0
1964	274 6
1965	283 4

(457).

ARPA requested \$277 million for FY1966, but was required to make more humble requests, totaling only \$254.1 million, for FY68 (458), and lower ones of, \$212 million, for FY1969. These amounts continued to decline through 1973, when they began to rise again. During the highest point of the ground war in Vietnam, at the turn of 1969 and 1970, ARPA funds were at their lowest, although as we shall see IPTO funds for AI did not change much. Paradoxically, the several tortuous years during which the war began to end saw the lowest ARPA allocation ever followed by an increase in funds starting in 1974. The 1971 budget request for ARPA totaled \$222.7 million, but the Agency did not receive this, and instead in FY1972 and FY1973 was given \$209.89 million and \$199.743 million respectively (459). In 1973, the Agency had requested \$226.727 million. The disparity between the request and the allocation of monies is significant, and this perhaps helps to explain the perceived miserliness of which ARPA administrators have been accused. This attitude must have indeed reflected the statutory stipulations of the Mansfield Amendment and the intense pressure from above placed on the administrators more than any change in sentiment. Total DARPA expenditures remained steady at about \$400 million in 1975, 1976, and 1977. Starting in 1977, a new category of spending called Major Demonstrations make up a significant increase in DARPA allocations. Long Term Technology Research Projects, as everything else DARPA did was called,

remained the same through 1980 and then started to increase (460).

The final layer of the onion represents actual expenditures on AI, usually under the auspices of IPTO or, later, Intelligent Systems and C3. These have been quite modest historically, usually totaling less than ten percent of the ARPA or DARPA budget. Invariably, some of DARPA's other concerns raised a good deal more money. When the FY1966 request for IPTO was \$15 million, \$127 million was requested for ballistic missile defense and \$59.3 million for nuclear test detection, out of a total of \$277 million for FY1966 for ARPA (461). During this decade, allocations for IPTO were fairly steady. IP technology was an \$18.8 million request for FY1968 out of a total \$254.1 (462). They rose from \$15 million (requested for FY1966) to the \$19.649 million actually appropriated for FY1969. FY 1968 was almost the same amount (463). However, the Department of Defense Appropriation Bill for FY1969 indicates that the amount for IP was \$25.950 million (464). ARPA 6.1 (that is research) funds was 30% less than in FY1970- exactly vindicating the statement by John McCarthy. Appropriations for 1971 and 1972 for IPT (6.1) were \$16.2 or \$16.1 million (465).

Shortly after this low point, DARPA's leaders began to dramatically restate the significance of ARPA's contributions to the "national technology base". They began to present the computing and AI programs as a defense application rather than simply indicating the wonders of the computer. Indicating the wonders of the computer, and even implementing timesharing and the ARPAnet for military purposes as some of the first testbeds, did not seem to be sufficient to continue to win significant funds. The greatly increased sums for ARPA itself were typically for exploratory development and strategic technology projects, some of which were applied AI. As we indicated in the previous pages concerning Dr. Heilmeyer, there was a threat-never exercised- to funding and thus there was an actual need for such re-orientation. The novel rhetorical base started with exactly the same line as ever: the USSR was a threat, and its aggressive spending could only be met with an equal U.S. response. In 1977, Heilmeyer told a Senate Committee that he was "somewhat overwhelmed by the sheer magnitude of the Soviet effort" (466). The response to this was the same as ever- continued U.S.

development of weaponry. But the tone was different in that it was increasingly infused with computing and other technology initiatives, and in that sense, the role of DARPA was different and the post-Vietnam military ethos was also different.

Heilmeyer, and others, recast arms issues as technology issues:

" When I appeared before this committee last year, I outlined an investment strategy which focused on some key questions whose answers are deeply rooted in advanced technology... These questions could become the national security issues of the 1980s...

" What are the technological initiatives in the command and control area that could enable us to use our current forces more effectively ? For example, can packet switching, intelligent terminals, or computer based decision aids significantly improve command and control ?

"...as a result of DARPA initiatives, while difficult technical problems remain, the technologies to answer each of these questions in the affirmative are on the horizon today..." (467)

As we noted earlier, the original administrative nature of the IPTO appears to have been a funding source for Licklider's former colleagues. Like the ONR before it, the early IPTO gave money to people its program directors believed were smart. Like many simple ideas, this one worked. This altered somewhat by the late 1960s, when the IPTO had formal criteria for different areas of technology, specifically automatic programming, picture processing, intelligent systems, and speech understanding. In 1977, the 6.2 designations Advanced Network Concepts and Advanced Digital Structures were added to the existing 6.1 categories.

FN x: Spending on these was as follows:

Dr. Lukasik: The total commitments for FY71 and FY72 were 16.2 and 15.6 million respectively."... [then gives specifics broken down into];

automatic programming FY71 522; FY72 2,455,000;

picture processing FY71 1,590,000; FY72 2,252,000;

intelligent systems FY71 5,254,000; FY72 6,016,000;

speech understanding FY71 790; FY72 3,537,000;

completed projects FY71 8044; FY72 1,340,000;

total FY71 16,200,000; FY72 15,600,000; (p774);

" The total requested for information processing tech inc 6.040 million for "intelligent systems".

Dr. Lukasik: The intelligent systems subproject was formed in FY1969. During the 4 year period FY 1969 1972, 19.6 million has been committed under this subproject." (p779);

FY 1973 ACTUAL IPTO ALLOCATIONS:

Information processing techniques

automatic programming 3160

picture processing 2507

intelligent systems 4534

speech understanding 3694

FY1974 ESTIMATE IPTO ALLOCATIONS;

Information processing techniques

automatic programming 2689

picture processing 2711

intelligent systems 3538

speech understanding 3762

FY1975 ESTIMATE IPTO ALLOCATIONS

Information processing techniques

automatic programming 3560

picture processing 2813

intelligent systems 4437

speech understanding 3480

APPROPRIATIONS FOR FY1975; Hearings before a Subcommittee of the Committee on Appropriations, House of Representatives. 93rd Congress, Second Session. Department of Defense Appropriations for 1975. Part 4, RDT&E. p2 summary by defense agency;

Computer and Communications Sciences;

ADD 000 TO ALL

FY1976 ACTUAL: 15 619

FY1977 ESTIMATE: 16 755

FY1978 ESTIMATE: 18 800

FY1979 ESTIMATE: 22 300

Image Understanding:

FY1976 ACTUAL: 3252

FY1977 ESTIMATE: 3212

FY1978 ESTIMATE: 3232

FY1979 ESTIMATE: 3350;

Intelligent Systems:

FY1976 ACTUAL: 8020;

FY1977 ESTIMATE: 9092;

FY1978 ESTIMATE: 8346;

FY1979 ESTIMATE: 6800;

Advanced Network Concepts:

FY1976 ACTUAL: -- not given;

FY1977 ESTIMATE: 717;

FY1978 ESTIMATE: 2537;

FY1979 ESTIMATE: 5350;

Advanced Digital Structures:

FY1976ACTUAL: 4347;

FY1977ESTIMATE: 3734;

FY1978ESTIMATE: 4685;

FY1979ESTIMATE: 6800;(p70);

Hearingsbeforea Subcommitteeof theCommitteeon Appropriations,UnitedStatesSenate.95th Congress,FirstsessiononHR 7933.Departmentof DefenseAppropriationsfor FY1978.February 4, 1977.77S181-27.1Part5, RDT&E. Statementsubmittedby Dr GeorgeH. Heilmeier.44-172.

With the pressure on military funding in 1969 and 1970, and with the statutory changes of the Mansfield Amendment, both bureaucratic shuffling of research designations and substantive changes in projects took place. More research was placed under the auspices of 6.2, exploratory development, than 6.1, basic research. Thus the nature of ARPA was recast. Because changes in grants policies meant more stringent criteria for funding, this was not universally popular- some people, like Nils Nilsson, “did not take redirecting well”. This period marked a transition into a more technologically oriented armed forces itself. This meant more money for AI; it also meant that the defense applications of that money had to indeed always be clearly extant. Such rationales had not always been clearly stated before, despite the fact that one can rather simply find defense rationales for all of this technology. This period, in turn, segued smoothly into the next period, that of the Strategic Computing Initiative (SCI) in the 1980s.

Changes in the Viscosity of the Flow of Research Funds

Changes in administrative policy arose not only in response to Heilmeier’s personal wish for closer scrutiny or a higher governmental attitude that pushed toward such scrutiny, but because of the new nature of the government-scientific research environment itself. In addition to the Mansfield Amendment requirement that research proposals indicate a clear military application, the funding itself was placed on a far more closely delineated basis. The multi-year projects were replaced, permanently, by year-to-year projects, which were subject to regular reviews (468). The change from tenured repose to a job which was subject to review also meant that it was simply much harder to administer research, and to ultimately sit down at one’s desk and get down to one’s work. According to ARPAnet pioneer Vincent Cerf:

“ I was finding it hard to get any research done because I was so busy...writing proposals, or writing reports telling people what I would do if I wasn't spending all my time telling them what I was going to do.” (469)

There was, as we shall see in a moment, not less money, but rather slower money. By this we mean a more tedious and competitive process of awarding grants for scientific research. But AI was also being practiced at more institutions, and a judicious dispensation of research money simply made a slower, more formal process necessary. A flood of entrants into a scientific field indicates success, and this may have to be paid for with bureaucracy. As more people entered AI, and further research programs were established, the field became too large to allow the awarding of grants based on informal phone calls between old grad-school pals. By the mid-1970s, a more formal competitive process was needed simply to assure justice. The formality may be easily accommodated with the fairness rationale, but combined with substantive requirements of ‘relevance’ and yearly or even quarterly reviews, it made research administration more difficult. Consequently, the bureaucracy surrounding such awards moved more slowly, even for those who have won such grants many times before. According to Michael Dertouzos:

“ Now it takes a year, even for a place like MIT to get a proposal through. The freedom of going down there with a quick idea and getting going is no longer there, because now there are all these bureaucracies set up, in the interest of fairness, for small people and big people to compete on equal terms -- all of which has a lot of validity. But then, of course, who says that you've got to be fair if you have got to be inventive. Then there is this tendency, which started for no good reason, which I call the systematic broadening and the systematic destruction of the strengths of DARPA, which was to take what was great and worked, and start chopping it up based on absolutely no rationale. So now you find three hundred universities competing, or three hundred contractors competing, in inky-dinky chunks.(470)

Institutional Expansion in the 1970s

Seeing DARPA troubles in such a dramatic light, as some researchers and administrators did, is itself a terribly near-sighted philosophy. First, what happened in the early 1970s was the portent of worse to come. The hold placed on the blank check customarily written to the field of AI in this time period was a preface to a stop payment written on that check after the Cold War itself ended. But this period was actually quite fortuitous in some terms, specifically that of commercial funding for AI. Although this technology was still quite far from any fielded applications, let alone polished products, the first commercial sponsors not expressly dependent on military contracting took the bait around 1970. As civilian commercial computing became a far more sophisticated business, AI inevitably piqued the interests of large computing-related corporations. The roster of potential rich uncles for AI itself gained some names.

Most notable for several reasons is the formation of PARC. Xerox Corporation opened the first commercial think tank, or rather, non-Cold War-related think tank, Xerox PARC, near the Stanford campus in Palo Alto in 1969. Quite early in its history, PARC brought in ARPA's Larry Roberts to be a major administrator; Daniel Bobrow was recruited to PARC (from BBN) as soon as it opened. The administrative similarities between IPTO and PARC led Allen Newell to call PARC simply an extrusion of ARPA (471), and true to this parentage, PARC proved consistently generative over decades. Ohio State University established its AI programs late in the 1960s. In 1971, the Information Sciences Institute was founded, in scenic Marina Del Rey, in Los Angeles, at the University of Southern California. Finally, we should note that the MIT Media Laboratory commenced its protracted gestation during the early 1970s (472). The Media Lab began as a series of mock-ups- suggestions of technical projects, without the technological back-up- by architect Nicholas Negroponte (473), once the project acquired corporate funding and a technical foundation an extraordinary amount of AI emerged from the lab. Thus, even its very beginnings were a fine omen for AI.

Chapter 9. The AI Culture Wars: the War Inside AI and Academia

As an heir to the philosophical tradition of empiricism, AI has been subject to extremely vituperative and harsh criticism. The practitioners of AI seem to be less affected- that is, less agitated- over this now than they were in the Sixties and Seventies, but the field has indeed been buffeted by the winds of scientific controversy.

In this chapter, we will consider objections to the central theory and practice of AI, and the field's own intellectual politics from its inception through the mid-1970s. In Part I, we will consider the relatively superficial criticisms that AI faced in the early part of its oeuvre. Career critics H. and S. Dreyfus have indeed made the field questionable in the view of the reading public (Part 2). Much as one hates to admit it, certain of these more serious criticisms have been taken to heart as valid by AI's practitioners. On the other hand, others within the field or close to it, such as computer scientist Joseph Weizenbaum, have become disheartened and critical (Part 3). But AI has not been only on the receiving end of injurious scientific politics. Several people within the core of the field of AI have criticized other branches of the sciences of the mind, specifically connectionist or Cybernetic theories of emulation of intelligence (Part 4). Finally (Part 5), we will see that at the beginning of the 1970s, it was the optimism of the AI people and NASA, rather than the pessimism of the critics, that triumphed eventually. But these viable successes led to their own problems (474).

AI's Philosophical Underpinnings, Revisited

The gut truth of Western philosophy, as both its friends and foes agree, is the coherence of the world and of the mind that perceives it. The Empiricists Hume and Locke may have disagreed with Rationalist Descartes on many things, but the sublimity of the mind was not one of them. Classical AI subscribed to both the Cartesian fascination with the mind and the Empiricist trust of the evidence of the senses. Certainly during the first years of AI, some of the critics may have been right that there was an unvarnished trust in the potential of science, AI included, to uncover the truth and build great things.

Such high aspirations are all well and good, but hubris is dangerous if not tempered by gravity and humility. The high and mighty must publicly profess humility today, and must contend with sexual harassment lawsuits, endlessly nosy news media reporters, dumpster-

diving private investigators, Environmental Protection Reports, talkative household employees, whistleblowers, webcams, malpractice insurance, Special Prosecutors, and the like. But none of these post-60s artifacts existed in the 1950s. The trait of humility was certainly not markedly evident among the scientific and political elite in the Postwar decade. While AI practitioners have been nose-to-the-grindstone scientists, the support of the field by the powers that be meant that those critical of 'power sciences' criticized AI as well.

The 1950s was one of the highest points of Classicism in Western civilization, and this theory merged seamlessly into the belief in American superiority- never mind the loose ends. Communism was uniformly denounced as a scourge in the mass media. American high culture became more adept at the Modernist fine arts, rather than only a weak would-be Europe. Every river was dammed and every bridge built (this was well before the conservation movement became a widely popular one). Evil childhood diseases such as polio were in large part conquered, in North America and Europe and increasingly elsewhere in the world. Smallpox was entirely eradicated in the early 1970s. The very conquest of outer space itself was taken on and accomplished; the first human walked on the Moon on July 20, 1969.

If AI carried with it some of the hubris which abounded at the time, this is no surprise. But even amidst the delirium of science and high culture at the time, AI set a high water mark. By setting its metaphysical grasp and its technological sights higher than anyone else, AI inevitably established itself as a lightning rod to which friction would flow. Moreover, insofar as the near-belief in omniscience in western culture and science was itself attacked, so too would the audacity of AI be attacked.

The mere logistics of the situation suggests a certain set of friends, and a certain set of enemies. The friends included the cognitive psychologists who worked with Newell and Simon almost from the moment they got started, and who both men cultivated at the RAND Corporation summer sessions and wrote with at CMU. Simon also worked with economists, but they did not take an interest in AI until much later. One would expect engineers and scientists to manifest enthusiasm for AI, but this was not uniformly the case. As we mentioned earlier, Norbert Wiener was struck with self-doubt as to his

contributions to implements of destruction, and his lobbying against postwar science included general complaints about AI (we will mention him again in a few pages). A number of engineers, among them Rand mathematician Richard Bellman and British engineer Stafford Beer, attacked AI as fraught with hubris and the wrong way to go about emulating intelligence. Newell and Simon's Rand colleague J.C. Shaw collected anti-thinking machine quotations. These statements ranged to the very extreme, for instance, as the Beer referred to computers as "less than morons" (475).

One would anticipate that certain other cohorts would be far more skeptical, simply by nature of their professional enterprises. A prospective list of AI skeptics might include humanists and artists, and especially philosophers. Some of this was sheer professional jealousy. AI addresses many questions of philosophy, such as the nature of learning and perception, in a way that is bound to attract more attention than philosophy does. This is the same mundane reason that the humanist often considers the engineer uncouth. The aggrieved party rushes to assert their importance, while nursing their indignation that universities don't pay them as well. The objections are more profound and personal as well. Engineers are not putting humanists out of business, since engineering technology does not make the study of the humanities irrelevant. But the relation between AI and philosophy differs here. If AI fulfills the hopes of philosophy by turning its speculations into engineering, it also supersedes the claims of philosophy by literally turning philosophy into an empirical science. AI is an engineering field, which entails doing something rather than simply arguing about things one can't prove. Once that is done, what will philosophers have left to do ? As we will see, this quandary was not lost on certain philosophers. These arguments are, for the most part, easily defused. This would not be true of the less superficial and thoughtless arguments forthcoming in the next decade.

2. Attacks on AI

"Alchemy and AI", un soweiter

Intensely divergent intellectual movements based on the same issues may coexist, or rather repeatedly confront each other over a period of years. Indeed, it is a likelihood rather than a conundrum that they live

at the same time. They are likely to take on this form as Hegelian opposites because of their common, if divergent, responses to historical circumstances. Classicism and its equal and opposite reaction, Romanticism, appeared in the guises of AI and phenomenology during the middle of the Twentieth century. A philosophy that radically devalues the veracity of the learning of the intellect, Phenomenology instead throws all of its chips into the qualia of individual experience. This experience can purportedly neither be understood formally, that is, through neurology, logic, or cognitive science, nor expressed in words between individuals. The early phase of AI coexisted with the heyday of phenomenological philosophy, and these two divergent ways of thinking have been invested with increasing vehemence as the century has worn on. It is one thesis of this book that science, in the case of AI tempered of the genre's worst excesses, appears to have won. But in the third quarter of the century, it was only a matter of time that they would meet in rancor. As is a Greek tragedy in which people are allegories as well as individuals, almost every bit of that acrimony appears to have been embodied, so to speak, in two brothers, Hubert L. and Stuart Dreyfus.

In the Spring of 1961, a distinguished panel, including Vannevar Bush, Claude Shannon, John McCarthy, Herbert Simon, Edward Feigenbaum, and others, presented a general discussion of the state of computing at a symposium at MIT. Two young philosophers in the audience, the brothers Hubert and Stuart Dreyfus, found themselves agitated afterwards at what they perceived as the outrageous arrogance of the computing projects, and apparently decided to do something about it. The two did not respond at the discussion following the session itself. But in a highly unusual move, they requested and received permission to insert into the published proceedings a half-page rebutting the work presented. A look at the text of the symposium itself underlines the sheer intensity of the animosity that Hubert and Stuart Dreyfus must have felt toward engineering in general (476). The symposium was hindered rather than helped by its blue-ribbon panel, for it called the elder statesmen of digital and even analogue computing to comment on the developments of the present. Unfortunately, some of their speeches indicate that they had not kept up with the state of the art in programming. John R. Pierce, the acoustics engineer who would oversee an influential and destructive report on machine translation several years later, claimed that advanced programming work including

machine translation, learning machines, game-playing and heuristic computing had not yet indicated their usefulness (477). John McCarthy and J.C.R. Licklider argued substantively that works in progress did not yet deserve such scorn, and that one had to be closely knowledgeable about programming- which Pierce was not- to comment. The Dreyfus brothers' post hoc commentary indicate that they had already made up their minds about computing:

“ After hearing Dr. Pierce's remarks attacked by the computing fraternity's vociferous far left, we feel that a few comments are in order. We undertake this defense primarily to show that MIT has not yet been taken over entirely by machines...”

“ Specifically we should like to address ourselves to the frequent and rather immodest claim of manifold progress in Artificial Intelligence. How much progress has been made toward the world champion chess machine (so confidently predicted) when, in the past few years, the rules of chess and a few simple-minded heuristics of its play have been committed to the punched card ? How much progress did the cave man make toward space flight when he climbed his first mountain ? Something fundamental to significant progress is lacking in both cases- the conceptual or technical or technological breakthrough. Work in the fields of language translation, game playing and pattern recognition has contributed nothing to the understanding of the nature of intelligence or insight. This suggests that the solution of profound problems touching upon the nature of thought may not be the sum of many minute steps.”

“ If programmers set themselves glamorous tasks as difficult to execute as they are appealing to the press (which includes some so-called technical journals), let them report progress when the necessary breakthrough occurs, and not until then. Such researchers should run the same risks as the alchemist trying to synthesize gold from base materials: obscurity until success. Artificial Intelligence seems to be operating instead on the principles of fame until failure. In contrast to Dr. Pierce's sober remarks, the claims of the far left read like fiction, and bad fiction at that, since the ending is always 'deus ex machina.'” (478)

The two would later assert that the acrimony started several years after this first assault. Even a passing knowledge of their backgrounds suggests that suspicion toward computers had been brewing for a good amount of time. Given the intensity of his sentiment toward rationality, it would seem to be only a matter of time before Hubert Dreyfus in particular would have found an object of rationality that became an itch that he could not stop scratching. Hubert Dreyfus had started out as a physics student, and then switched to a career in philosophy, finishing his degree at Harvard and then moving to teach at MIT. He openly states that he made his move in graduate school because of “the excessive rationality of physics” (479). He was teaching philosophy at MIT at the time of the presentations, and had heard of Minsky’s ‘Robot Project’. He said this was presented to him by the students as having “solved the problems that philosophy worried about, like understanding and knowing and so on” (480). Stuart Dreyfus was research associate to the distinguished Rand Corporation mathematician Richard Bellman, and later he worked at Rand and helped his brother get a job there (481). While he has collaborated in some of his brother’s published work, he has generally been less involved. This was the first public act against AI by Hubert Dreyfus, who later moved to U.C. Berkeley, where he has written several books and many articles attesting to the perfidies of AI.

Hubert had found his calling and proceeded with energy. With his brother’s introduction, he presented himself to Paul Armer, then director of Computing at the Rand Corporation, as a disinterested philosopher. Armer gave him a job for the Summer of 1964. Later on he asserted that he thought he might bring in a philosopher just to introduce some new ideas into the environment. He also stated later that Dreyfus did not honestly inform him that he already possessed heated beliefs about AI and that these results had been published already (482). Hubert Dreyfus claims in the 1991 Baumgartner and Payr interview that he did not hold strong opinions on AI when he arrived at Rand. In Mind Over Machine (1985) the brothers assert that RAND recruited Hubert to look into the issue of AI (483).

Of course, an impartial dialogue is not what resulted. When Armer saw the paper which resulted from the summer’s work, he found it to be “vindictive and angry and poorly written”. He was adequately alarmed to delay its publication for nearly the entire academic year (484). But

"Alchemy and AI" was out of the bag by mid-1965, as Rand Corporation 1965, P-3244) (485).

"Alchemy and AI", which Dreyfus asserts is one of RAND's best-selling research papers ever (ibid.), is the most extensive, that is, lengthy, critique of the AI program ever. Its arguments have been returned to print so many times over four decades that it has become the best-known by a good margin. "Alchemy and AI" was eventually turned into a book, What Computers Can't Do ('WCCD,' 1972; Revised Edition 1979; Second Edition with new introduction, 1992) and other works (Mind over Machine, 1985) which are essentially revisions of the first (486). The essence of the argument has remained unchanged, as the core of the books continues to be a discussion of AI research from 1957 through 1967, and of the "assumptions" behind AI's "continuing optimism" (487).

In addressing the work of Hubert Dreyfus on AI, it is hard to know where to begin. Seymour Papert devoted a good chunk of time in the middle of the 1960s to writing a fine debunking of Dreyfus' 'debunking' of AI. But later he admitted to regretting throwing good energy after bad. However, Dreyfus has succeeded in becoming, oddly enough, perhaps the most famous figure associated with AI for the general public simply by throwing his case in front of the media so often. Thus one may not have a choice but to confront him (488).

The attack is essentially simple. In each of the four presentations of "Alchemy and AI" (the original document, the 1972 book, its 1979 revision and the 1992 revision, among others), he attacks AI for dozens or hundreds of pages and then presents his alternative proposed understanding of intelligence. Curiously, the seed of the proposed alternative means of understanding the mind, is buried at the back of WCCD, starting only on page 231. It might be a better idea to start there and then consider the rhetorical and substantive excesses of his attack on AI.

The basic division in Western philosophy, as we have sketched it in the duration of this book, is between the rationalist distrust of the senses in opposition to the mind (Kant and Leibniz, and Cartesian or 'Continental' philosophy), and the empiricist trust of the basis of the mind in earlier sense-perception (Hume, Locke and English Empiricists). But the 20th

century vogue of phenomenology offers a further and very distinct view, holding that qualia- felt experience as opposed to obviously measurable things- are foremost (489). Sense experience is evidently available to animals as well: human beings are conscious because they have something more. 'Intentionality', the special human quality, is the ability to think about something rather than simply what is in front of their noses.

One could say that the emphasis on intentionality simply points to the importance of human abstraction and cognition, and simply leads us back to Classicism of either a Rationalist or Empiricist form. But still, these are points well taken- and they have indeed been taken seriously by AI too (as in the movement to appreciate non-cogitational forms of intelligence, e.g., Brooks). In its simpler form, phenomenology should be seen as a good antidote to too much logical positivism, that is too much belief in the rational and the demonstrable. This also existed in a too-piquant form in mid-century as well.

But in the form that phenomenology has taken, ideas are carried to an extreme which is nearly impossible to defend. Martin Heidegger (1889-1976), the nearly unreadable German philosopher who Dreyfus cites so often, took these ideas and turned them into something much more politically and socially forceful than an assertion that perception per se is complex and an object worthy of study (490).

Specifically, Heidegger turned early psychological Phenomenological observations into an extreme Romanticist exposition on the alienation of modern man from Nature and the so-called 'oneness of the world'. He was also the Rektor (provost) of the University of Freiburg beginning in 1933, and in an infamous inaugural address openly expressed the University's support for the Third Reich. This author brings this in not only because it is an intellectual hot poker, but because it helps to indicate that calls for a return to Nature and rejection of rationality may be just as dangerous as explicit claims of the superiority of reasonable and rational civilizations. In this case, the "oneness with Nature" bunk was applied to Nazi purposes: Jews and other non-Aryans were specifically accused of lack of "clarity and understanding of nature". (491)

The concept of “restoring the oneness of man with nature” is so intrinsically radical that one might imagine that by its very nature it could be twisted to scurrilous ends. And indeed, it has been. Heidegger’s association with Nazism is one from which it would seem to be impossible for anyone to ever exonerate themselves. It is the same sort of simplistic Romanticism that ushered in the reign of the Jacobins after the French Revolution. Moreover, it would seem to indicate something very fishy about his other ideas. One could say this about Christianity as well, for a good reason: restoring the oneness of man with nature, like asserting a new moral order, is a very strong statement which may be used to justify many questionable acts. AI has always been taken up with an explicitly scientific rather than moral or normative agenda on the plate, so it would seem that it could escape this charge of potentially dangerous ideas.

Phenomenology was actually reappearing at about the time that Dreyfus and Dreyfus were getting started. In all sincerity, it must be said that phenomenology as Dreyfus undertook it was a serious, articulate American wing of European high culture. Later, it was adopted into youth culture with a vengeance and turned into a soggy mulch of pop culture celebrating “experience”, and inspired innumerable bad drug trips and drunken car accidents. Still, prevailing sentiment at the time being on his side, he took advantage of the historical moment.

The usages of phenomenology by Hubert Dreyfus are apparently non-political and more innocuous, if no more intellectually meritorious: he wishes to harry and hound AI, and has done so for decades. He believes that things are not to be taken in any abstract sense- that there is no reality apart from felt experience, which in itself is a highly unqualifiable qualium. This strong statement has further implications for knowledge per se, although Dreyfus has customarily gone after AI rather than any other purportedly equally invalid social sciences. Pamela McCorduck tells us that:

“ He agreed that his alternative cannot on principle offer any kind of scientific understanding of these things, that in fact he believes that human intelligent behavior is a sort of something or other which we cannot have a scientific theory of. Thus all social sciences are for Dreyfus, as wrong headed as AI. This is not an attitude widely held in universities.” (492)

This means that one cannot have a problem solver, much less a general problem solver, because it cannot be put in context as human problem solving is in context. That is, human beings have sensory means by which to taken in information; they can also translate thoughts into actions (493). In robotics parlance, humans have receptors and effectors. We have bodies which are necessary for both of the latter (494). But equally important, human perception is not without context: it has what is known as intentionality (495). This philosophical concept ties together abstract 'declarative' knowledge (or 'book learning', as folk psychology calls it), with embodiment. This means the context, either physical strictly speaking or circumstantial, in which something takes place (496). Learning is so deeply embedded in particular social and personal contexts that trying to remove it for any disembodied task will only succeed in a trivial way.

Embodied intelligence and the intentionality of human conscious states are complemented by one other salient feature which HD tells us is the sine qua non for humans, and an impossibility for computers. This is fringe consciousness, a physiological and intellectual feature of humans (497). Just like a picture which must be put in some sort of structure or frame before it can be hung on the wall and displayed, human intentionality requires that one focus the eyes and the mind by placing a frame (or a 'fringe') on an object before one focuses on it. One can concentrate on the items in the visual field because one can sense the edge between the object and where it ends. The photographer knows to establish a visual gap between subject matter and the edge of the photo, and the computer user knows to turn the contrast up to avoid wearing out the eyes. But these are conscious acts of will, which only humans can carry out, we are told, and which computers cannot because they lack consciousness per se.

The goal of general, human-like intelligence carried out by a computer is patently impossible, says Hubert Dreyfus in his publications, because computers lack these qualities of embodied consciousness and hence intentional thought. The features of human intelligence is such that it cannot be recreated at all mechanically. In order to do this, indeed, it would have to distinguish the relevant from the irrelevant, perceive things that are understood almost subliminally (i.e., "fringe consciousness"), understand the context in which a given datum is to be understood, and generalize from individual cases to classes (498).

Computers, he says, are particularly unable to engage in these more subtle skills of distinguishing the trivial from the urgent and placing items in the proper setting.

A good deal of this seems like perspicacious folk psychology, rather than a particular refutation of AI. Indeed, some of the observations, such as the frame structure, to define the context in which certain actions take place, have been adopted into the innovations of AI. But they have not been adopted because of Hubert Dreyfus. The manner in which he uses cryptic and bombastic philosophical observations suggests a born-again Christian rather than a disinterested scholar. Consideration of Dreyfus' argument suggests overly enthusiastic usage of rhetoric. Like Heidegger's embrace of political causes which later caused the death of fifty million people, Dreyfus' embrace of rhetoric casts a pall over his intellectual work. (As he himself says, things must be taken in their larger contexts). It is simply clear that Dreyfus is adequately well-informed to have taken a broader interpretation, but chooses the narrower and more condemning one.

Even an incomplete consideration indicates that these books ride roughshod over a great deal of intellectual territory. In the first and in subsequent works, the nature of AI is presented in a list of several fundamental assumptions- biological, epistemological, and ontological. Dreyfus alleges that the practitioners of AI (understood very broadly) allege the congruity between the digital states of a computer and the ostensibly binary states of information in the mind. This charge is on its face untrue- and it was even at the time of "Alchemy and AI." Moreover, the literature citations concerning the possible congruities is almost entirely from the 1950s: the theorem of binary state thresholds in the CNS had been dismissed by the time Frank Rosenblatt updated the Perceptron. However, Dreyfus does not correct or update the text of the 1972 book in later editions. The educated, but non-technical reader would not know this, of course (499).

Similar mishandling of the literature is the case with the "psychological assumption", in which Dreyfus tells us that AI believes human psychology to be as unsubtle as a filing cabinet full of manila folders (500). This is a remarkable indelicate way to address the highly subtle and thoughtful work of NSS and their colleagues in cognitive science, whose path of inquiry iterated between cognitive psychology with

human subjects and cognitive modeling with computer programs. Moreover, someone who believes that all social science is bunk is destined to condemn AI.

Most of these arguments are nothing to get too worried about, in large part because they are, or were already, objects of central concern to the field and to related fields even by the time What Computers Can't Do was first published. For instance, the central argument from Wittgenstein concerning the parallel "crunched-togetherness" of experience, had already been considered by Oliver Selfridge when he referred to consciousness as parallel and best embodied by numerous agents ('demons') rather than a grand central information processing node. The acquisition of "intuitive" understanding of well-learned tasks was being studied by cognitive psychologists such as Anderson and Bowers 1973 (501). Tasks which we know so well that we perform them without conscious attention are referred to as "compiled", or deployed with automaticity (502). Both the frame problem, considered later in this chapter, and Frames, one of Minsky's salient contributions, acknowledge that making the distinction between major and minor issues, and placing items in the proper context, require much real-world knowledge. Moreover, both of the latter clearly admit that AI would have to construct common-sense knowledge and the ability to discriminate. More recently, other features of human cogitation have come under scientific scrutiny. As we shall see later, many people in AI have conceded the need for a body in order to have intelligence of the anthropomorphic sort. (In order for AI to get a life, it will be necessary to get a body first). The attack on AI thus does aim at the most serious concerns, but in such an angry and vindictive way that it is hard to see WCCD as constructive.

The most ferocious enemy is a desperate one- one gets that sense with Dreyfus. This would account for the intensity of his ideas. It is interesting to note that Dreyfus' boundary of impossibility has been moving forward slowly as AI itself has progressed. He first claimed that computer programs could never play chess as well as a ten-year-old, and then had to take that back when one such program beat him in 1968 (503). He indicated that micro-worlds were merely that- toy domains- without ever addressing the broader emulation of decision making in production systems and their subsequent development by Newell and many colleagues at CMU. By 1979, he was acknowledging

that the field of knowledge engineering could produce useful artifacts, contradicting earlier assertions that AI would not even produce chess-playing programs. By 1992, he ignores AI's achievements and focuses on Lenat and Feigenbaum, who do not compose the entirety of the field. He asserts that AI people would not face their own weaknesses, without ever conceding his own. He neglects the field's longtime concern with common-sense, and the field's own self-examination with the introduction of the frame problem and frames and schemas as forms of knowledge representation. He claims that the 'Shakey the Robot' project at SRI was dismissed because of its own flaws. This is not the case: the project was shut down by DARPA during an interval of dramatic budget cuts. We must gather that Dreyfus knew this and declines to mention it (504). The list goes on.

Predictably, the AI community has responded by making him persona non grata, combined with wary attention to his arguments. Amazingly, Dreyfus took this personally, saying that people were frightened to be seen in his presence: "the rejection was so total that students and professors working on the robot project dared not be seen having lunch with me without risking getting into trouble with their superiors." Apparently, there was no dictum from on high (505). Various AI people have debated him in public over a number of years, however, and he has participated in forums and roundtables at AI gatherings. Minsky and Papert engaged in the debates for years, following a change of cast when Dreyfus moved to perpetually contentious Berkeley (506).

Dreyfus asserts that this refusal to accept the phenomenological claims regarding the holistic nature of experience is a coward's refusal to defend his or her weak and flawed thinking (that is, you don't agree with me, so you're a fool), specifically, the narrowing of debate necessary to a failed and decaying research program. However, his attacks have been, historically, so angry that it seems more likely that many of AI's practitioners were loath to engage him. Edward Feigenbaum berated Dreyfus for offering AI the "ball of fluff" and "cotton candy" of phenomenology, and says that we need "a good Dreyfus" (507). Several "good Dreyfuses"- most notably Paul and Patricia Churchland, and Daniel Dennett- later emerged from the field of philosophy.

3. The Insider as Outsider

Insiders can become outsiders by treating themselves as such. AI has not been particularly arbitrary or exclusive for anyone who has managed to get to participate- it has even been relatively lacking in rancor for Dreyfus. Joseph Weizenbaum, an MIT electrical engineer who became angry with the field, was not the first insider to try to become an outsider. That designation belongs to Yehoshua Bar-Hillel, was one of the first people to work on machine translation. He later repudiated his work on this, and turned to arguing with John McCarthy and others at conference discussion sessions and in print (at the Uttley conference and in Greenberger ed, 1962). Acoustical engineer John Pierce was also highly suspicious of AI (508), and was one of the people most responsible for the ending of funds to the early research in machine translation more specifically in 1965. Aiken Laboratory machine translation researcher Anthony Oettinger, who found his work unfunded at the end of the 1960s, also became angry at AI and actually wrote the introduction to What Computers Can't Do.

MIT professor emeritus of electrical engineering Joseph Weizenbaum (1923-2008), distinguished for his work in early timesharing projects, remains the most salient example of an AI insider who behaved like an outsider (509). The author addressed his critiques of the field as such. Many of his shots at Artificial Intelligence research simply distill into an attack on the character of AI's participants. This is unfortunate because certain other of his arguments must be taken seriously.

Weizenbaum learned about computers the hard way: he built one when he was a graduate student in the 1950s. For want of a commercially available digital computer, a math professor drafted Weizenbaum and others to build a computer for the university (510). As a result of this apprenticeship, he was hired in the late 1950s as a consultant to the Bank of America's joint project with SRI in the computerized processing of checks. During this project, he became acquainted with list processing, as well as with researchers at Stanford's Computation Center.

In 1963, Weizenbaum started at MIT as an assistant professor of electrical engineering. It was here that his disenchantment with the extremes of human-computer interaction set in. The timesharing-mediated interaction with computers was taken far too seriously, he

believed. He decided, in response, to show what he perceived as the superficiality of knowledge representation, a field which he freely admits he was neither educated nor interested in. But he was a programmer. Apparent NLU which does not truly constitute semantic and syntactic 'understanding', is indeed easy to parody. The Eliza program, named after Eliza Doolittle in Pygmalion, did exactly this (511).

Weizenbaum created this program in an attempt to parody AI, and to show how easily AI could trick the gullible. Eliza purported to be an 'artificial' psychologist, which could interrogate a patient (512). However, the program's initial question "Why did you come here today?", and its responses to the answer to this question have no semantics whatsoever. The program is only a parser trained to find the direct object in the patient's statements and then suggest: "tell me about it", pick up the noun phrase in the answer to that and say, "why do you feel that way?" and periodically return to the topic. Eliza was readily tripped up by its sheer syntactic block-headedness. However, emulations of psychological states are not hopeless per se. Kenneth Colby, a psychiatrist and one of Weizenbaum's friends from Stanford, took the ostensible aim of the Eliza program at face value, soon afterward created a counterpart program, Parry. Parry was seriously intended to simulate a paranoid schizophrenic.

Weizenbaum was shocked to see that his prank was taken at face value:

" A number of practicing psychiatrists seriously believed the Doctor [i.e., Eliza] computer program could grow into a nearly completely automatic form of psychotherapy...I was startled to see how quickly and how very deeply people conversing with Doctor became emotionally involved with the computer and how unequivocally they anthropomorphized it." (513).

He cited a 1975 Natural History article, in which Carl Sagan suggested that Eliza could be implemented for affordable psychiatry for the masses (514). In "In praise of robots, Sagan gives the quotes from Eliza, then says that this should substitute for actual therapy:

“ This astonishing- one is tempted to say perceptive- response from the computer is of course preprogrammed. But then so are the responses of human psychotherapists. No such computer program is adequate for psychiatric use today, but the same can be remarked about some human psychotherapists. In a period when more and more people in our society seem to be in need of psychiatric counseling and when time sharing of computers is widespread, I can imagine the development of a network of computer psychotherapeutic terminals, something like arrays of large telephone booths, in which, for a few dollars a session, we would be able to talk with an attentive, tested and largely non-directive psychotherapist.” (515).

The experience of a bad joke taken seriously helped to lead Weizenbaum into a harsh condemnation of AI's precepts and (certain of) its people. In Computer Power and Human Reason (1976), his widely-read critique of AI and computing, Weizenbaum throws the baby out with the bathwater by putting down AI, rather than people's naive beliefs about it. Weizenbaum also took the condemning side in the 'culture wars' over AI. In this endeavor the Eliza program was one of the most useful 'straw men' for AI's detractors. The central distinction drawn in the work is that between autonomous and prosthetic technologies, the latter being tools that extend and enhance human sensory and cognitive capabilities. The former are tools that are more complex and which embody some internal model of the world- but which run in accord with their own rather than with human beings' 'natural' activities. The clock, to use the book's own example, replaces phenomenologically human experience with artificial, objective scientific time. (Weizenbaum cites cultural historian Lewis Mumford as the original source of this terminology). Direct human experience is, putatively, increasingly replaced by machine-mediated or 'artificial' experience. This is a very close analogy to Marx' discussion of the decreasing organic composition of capital. Marx contends that human labor, hand-crafts style, is the sole source of value in manufactured goods. Mass-produced goods are intrinsically worth less as such- German romanticism applied to economics. In both cases, it is important to see the philosophical roots of the ideas. Yet algorithmic rendition of the complexities of human perception, and especially the complexity and idiosyncrasy of human speech, is in itself at best a dubious quest. This argument leads Weizenbaum to the conclusion that Artificial Intelligence is an immoral scientific quest.

Weizenbaum asserts, to boot, that the cultural trappings of computer science themselves rob their participants of social and personal lives. This other aspect of his attack is much weaker. Weizenbaum alleges that many people in AI have psychiatric problems, and are so caught up in their work as to preclude the development of normal social skills. This assertion is roughly the same as Sherry Turkle's comments concerning hackers, made famous in The Second Self. It was been fiercely rebutted by Minsky, who sees the hackers as "the most sensitive, honorable people that ever lived" (516).

The Insider-Outsider's Moral Critique

Finally, Weizenbaum spoke of morality and responsibility. He stated that views of the mind as simply a biological organism which can be replicated are morally tantamount to Nazism. The usage of the term 'meat' rather than 'flesh', is a grotesque dehumanization of intelligence itself. Minsky has clearly used the phrase to provoke rather than in malice; we don't need to pursue the endless meshugas between Minsky and Weizenbaum over the years. But Weizenbaum's claim that this is a poor choice of words, and that AI should honor rather than demean the capacity of intelligence, is the most convincing of all of his arguments.

"...About thirty years ago, in an article in Life Magazine, - an interview, perhaps, Marvin Minsky made the now famous statement, which he has never taken back or modified (I heard him make it again last year; I think he is proud of it) that the brain is merely a meat machine. What is interesting and what perhaps escapes readers in other languages is that in English we have two words for what he calls "meat"; meat is dead, can be burned or thrown away, whereas flesh is living flesh and a certain sense of dignity is associated with it. We don't talk about eating flesh, and if we talk about burning flesh it is a horror image. Why did he say meat machine and not flesh machine ? It is a very deliberate choice of words that clearly testifies to a kind of disdain of the human being." (517).

Meat machines is certainly not a perspicacious choice of words, but such materialism is confounded by the fact that AI practitioners work on the capacity of the mind their whole lives. In a 1977 letter to his daughter Barbara, Herbert Simon indicated both the modesty and the magnitude of the AI initiative:

" I don't think we have to assert any special uniqueness of man, or separateness from the rest of nature in order to find value in life. Personally, I find it more

agreeable to think that man is a part of nature than that he is apart from and above it... Second I think those who object to my characterizing man as simple want somehow to retain a deep mystery at his core- a denial, again, of his integral relation with the rest of nature.... to show that something whose behavior looks very complex and erratic is really built from the combinatorics of very simple components is beautiful, not demeaning. It would seem to me than every scientist would have to think so, for the whole purpose of science is to find meaningful simplicity in the midst of disorderly complexity." (518).

Chapter 10. The AI Culture Wars: Popular Culture

"I'm sorry Dave. I'm afraid I can't do that." (519)

When phenomenology and its ilk moved from high culture, arcane philosophical circles into popular culture, specifically the youth culture of the moment, it wreaked havoc, pulling science further away from culture and art and evoking a more general distrust of science and technology. This chapter will consider extreme statements in the press as well as the most evocative spectre of AI ever created- the murderous HAL computer in Kubrick's 2001: A Space Odyssey.

1. Mishaps in the Press and Fear of the Future of Computers in the 1960s

The perspective of the early 1960s was such that the field had barely gotten started, much less finished. Its state was nearly gestational, and its very practitioners were not even reconciled to a common name or means to inquiry. Thus the nature of attacks against it and defenses against those attacks, differed from what they would be later. Attacks were often poorly thought-out and contemptuous, of the 'heads in the sand' sort that Turing mentions. But by the early 1970s, AI and fields that were close to it in spirit and sometimes in subject, had accomplishments to show off. These included SHRDLU and Shakey the Robot; whatever critics said, both were impressive and viscerally appealing. They had great value in terms of the 'wow' factor. Vision, robotics and early haptic and visual perceptual programs were also beginning to work. This changed the nature of attacks from the off-handed and shrill- "this is too far-fetched to ever be possible"- to the nervous- "this is morally outrageous and must be closely watched". There is a difference in the nature of attacks on AI visible from Taube and

Turing, which predate most actual AI work, versus those visible from sensationalist science journalism, epitomized by the 1970 Life magazine article. We will look at the latter, and then consider the real scientific watersheds which were breached at this time. Finally, the film 2001 practically counts as a scientific achievement in itself, and should be considered as an exposition of the possibilities of AI and as a presentation of the first Moon mission- a year before the fact.

This was a time of strong polarities: one person's reasonable extrapolations were (and are) another's wild dreams. It was inevitable that some people would see the very enterprise of AI as irresponsible exertion of power. While there is not much justification for such charges against the leaders of AI, the very idea was too good a target to not lob at. Luddites love to talk about scientific follies, the more prestigious, Promethean, and long-running, the better. Computer hype is not much different in its grandiosity from any technological utopianism- it's drearily similar, as are attacks on it. This is evidenced in an article, by journalist Bruce Darrach, which appeared in Life Magazine in 1970. The article began as a reasonable informal exposition of how Shakey the robot could follow simple directions in natural language, find the correct way to proceed up a ramp, etc. The generally even-handed article is quoted in Chapter Six. But then Darrach turns to a very au courant 1970-ish essay on the dangers of reliance on computers, on our society's disturbing reliance on information-processing.

"... AI experts believe that fiscal planners in both industry and government, caught up in deepening economic complexities, will gradually delegate to computers nearly complete control of the national and even the global economy. In the interests of efficiency, cost-cutting and speed of reaction, the DOD may well be forced more and more to surrender human direction of military policies to machines that plan strategy and tactics..." (520).

Never mind that no such thing as "computer direction of the national economy" or of nuclear defenses exists, and that major mishaps, such as Chernobyl and the Exxon Valdez, have typically been caused by sleep-deprived human beings, not evil computers. This presentation invariably implies that the computer will be some sort of God-like omniscient Big Brother, when in fact what has appeared are a multiplicity of independent agencies and their

associated computers. Far from being all-powerful and HAL-like, the left hand of one corporation often does not know what the right hand is doing. This hardly inspires anxiety on the level that Darrach seems to be trying to evoke.

The anxious vision of a world controlled by computers is followed by an odd presentation of Marvin Minsky's purported words as a kind of civilian Doctor Strangelove:

“ If a computer directing the nation's economy or its nuclear defenses ever rated its own efficiency above its ethical obligations, it could destroy man's social order- or destroy man. “ Once the computers get control, says Minsky, “ we might never get it back. We would survive at their sufferance. If we're lucky, they might decide to keep us as pets.”

“ But even if no such catastrophe were to occur, say the people at Project MAC, the development of a machine more intelligent than man will surely deal a severe shock to man's sense of his own worth. Even Shaky is disturbing... Is the human brain outmoded ? Has evolution in protoplasm been replaced by evolution in circuitry ?“

““ And why not,” Minsky replied when I recently asked him these questions. ‘After all the human brain is just a computer that happens to be made out of meat.’ “

“ I stared at him- he was smiling. This man I thought has lived too long in a subtle tangle of ideas and circuits. And yet men like Minsky are admirable, even heroic. They have struck out on a promethean adventure and you can tell by a kind of afterthought in their eyes that they are haunted by what they have done. It is the others who depress me, the lesser figures in the world of AI men who contemplate infinitesimal riddles of circuitry and never once look up from their work to wonder what effect it might have upon the world they scarcely live in.” (521)

Minsky responded that he did not say these words and that many of the quotes were fabricated. Life would not publish a retraction or his statement accusing the reporter of false statements. In 1997, Minsky was still being quoted, and was still denying that he had ever said this- “that Life quote was made up. You can tell it's a joke”. (522) A number of those who had taken time out to speak with Darrach and were now being called “lesser” colleagues and shown as soulless henchmen, felt hurt and betrayed. Minsky and Papert responded that Life should have refrained from talking about this topic in “a Frankenstein atmosphere”, but there was not much they could do about this.

Minsky and Papert replied eloquently in the Letters to the Editor.

“ Intelligence is a gift man scarcely understands. Knowing how it really works would change the world. As your article says, this knowledge might someday be used to build robots that imitate or surpass us. But even if society somehow decides to reserve intelligence for people, understanding our minds would surely change how we live, how we bring up our children, what we think we are.

“ What a pity Life chose to discuss these issues in a Frankenstein atmosphere and to make it seem as though Minsky had mysterious dealings with the Pentagon and believed scary robots were right around the corner. The aim of research on AI is not simply to construct robots but to develop computer models for testing psychological theories. Eventually this will change our ideas about the mind, as sailing the seas once changed geographical theories from philosophical speculation to practical science.” (523)

The statements of AI practitioners were for the immense majority restrained. But their temperance did not prove to be of any particular help, given the typically overwrought technological visions which counterpoised them. A volley generally aimed back at these statements, and at the Cybernetic and computer community would give AI a black eye as well. Throughout the post-war period, atomic scientists, cyberneticists and occasionally computer scientists had made untoward statements extolling the virtues of computing and the automation of manufacturing.

In 1975, Harvard press officer Fred Hapgood wrote in The Atlantic:

“ Every culture has its juvenile embarrassments; misdirected enthusiasms which fail dramatically and in retrospect seem to say something humiliating about the civilization that pursued them. The great computer craze of the late 1950s and the sixties is such a case. From the erecting of the machine, any number of respected thinkers derived a vision of society. Edward Teller foresaw an automatic world, ruled by machines. Gerard Pile, publisher of Scientific American, wrote and spoke about the “disemployment of the nervous system.” C.P. Snow thought that automation would be a revolution with effects far more intimate in the tone of our daily lives than either the agricultural transformation in Neolithic times or the early industrial revolution.’ ‘Is the handwriting on the wall for the labor movement ?, the Wall Street Journal asked, looking at the matter from its own perspective... The Ad Hoc Committee for the Triple Revolution (weaponry, automation, human rights), which was a study group composed of social luminaries like Gunnar Myrdal, Linus Pauling, A.J. Muste, Michael Harrington, Bayard Rustin, Irving Howe, Robert Heilbroner, and Tom Hayden and Todd Gitlin of SDS, saw the coming of automation as an argument

for a guaranteed minimum income. In 20 years, wrote Donald Michaels in a Center for the Study of Democratic Institutions book, most of our citizens will be unable to understand the cybernated world in which we live... the problems of government will be beyond the ken of even our college graduates. .. There will be a small, almost separate society of people in rapport with the advanced computers. These cyberneticians will have established a relationship with their machines that cannot be shared with the average man. Those with a talent for the work probably will have to develop it from childhood and will be trained as extensively as classical ballerinas." (524)

Even among the most respected scientists, this enthusiasm sometimes resulted in endorsements of computerized help in contexts in which it may still seem, decades later, to be most unneeded and out of place. When astronomer Carl Sagan endorsed the usage of robots in space exploration in a Natural History article in 1975, it seemed highly reasonable. When Sagan went on to suggest the mass deployment of computerized ELIZA-type therapy for those who could not afford the psychoanalytic Grand Tour, it struck some as far-fetched.

" This astonishing- one is tempted to say perceptive- response from the computer is of course preprogrammed. But then so are the responses of human psychotherapists. No such computer program is adequate for psychiatric use today, but the same can be remarked about some human psychotherapists. In a period when more and more people in our society seem to be in need of psychiatric counseling and when time sharing of computers is widespread, I can imagine the development of a network of computer psychotherapeutic terminals, something like arrays of large telephone booths, in which, for a few dollars a session, we would be able to talk w an attentive, tested and largely non-directive psychotherapist." (525).

Alternatively, the latter generation of the Cold Warriors and computer scientists was often much more restrained, as if the Vietnam War had cast a pall on the larger enterprise of science and technology. One could often witness introspection and doubt and even public breast-beating among scientists circa late 1960s. This extended beyond the very public recrimination of the nuclear cold warriors, which had been intermittently ongoing as early as the late 1940s (e.g., most famously, by Norbert Wiener and J. Robert Oppenheimer). (526)

It is interesting to see how much commentary turns to ether in the face of solid history. The words of Dreyfus and Dreyfus, and

other more elite and reserved people such as Oettinger, Pierce, and Bar-Hillel, as well as the words of the extreme technological utopians, were simply froth on the sea in the face of genuine progress. The universal antidote of reality simply made much of the skepticism seem sour and prissy. Elation seem to be in need of a glass of ice water. An ounce of implementation, it seems, is worth a pound of hand-waving. The actual commercial computing products which appeared during the late 1960s included the first appearance of electronic funds transfer EFT transfer systems; a dramatic decline in the cost of CPU power with the introduction of integrated circuits for ALU operations; the foundation of Intel and the invention of the first memory, control, and ALU on a single IC microchip; much dissemination of computer timesharing, and a generally increased implementation of business computing. Nothing succeeds like success.

2. AI in Pop Culture '2001: A SpaceOdyssey'

The Reality of 2001, circa 1968

The reality was more forceful than the movie: humans did indeed fly to and walk on the moon, on June 20, 1969, and then on subsequent occasions. But the movie 2001, released about a year earlier, was itself adequately meritorious as educated guesses about the future of science to warrant a consideration. It vindicated many bold assertions about technological possibilities. The dream helps to pull the reality forward, especially when the dream has decent foundations in reality. Moreover, the script itself was quite substantive. Director Stanley Kubrick had consulted about two dozen scientists for his script, and Arthur C. Clarke, the world's leading exponent of 'hard' science fiction for the book. This makes the movie a serious and thoughtful technological prognosis for its time as well as the other things that it is, rather than seeming retro, antiquated, or otherwise silly.

Part V. Big Ideas and Hardware Improvements in the 1970s

Chapter 11. AI at MIT in the 1970s: The Semantic Fallout of NLR and Vision

Introduction

As we saw in Chapter Four, the development of AI as cognition at the MIT AI Lab had been accompanied by the equally active development of AI as embodiment. A number of people worked in the receptor-effector tradition, but there were also those in classic AI asking questions such as 'How does search proceed', 'how does problem solving take place', and 'In what form are things stored in memory'- 'and what does the phrase stored in memory mean, anyway'. If the quick sketch of AI at MIT as characteristically bottom-up (starting from specific cognitive modalities) is occasionally the case, it is also proved to be too simple a sketch by the fact of lots of top-down (directly looking for general aspects of intelligence). Minsky and Papert indicated the bottom up creation of intelligence could not be successfully engaged without bringing in the 'top-down' as well:

“ ‘ Faced with the apparent diversity of kinds of knowledge, the common approach at [non-MIT] centers is to seek ways to render it more uniform so that very general logically clear methods can be used: the [MIT] approach is to accept the diversity of knowledge as real and inevitable, and find ways to manage diversity rather than eliminate it.’ ” M. Minsky and Seymour Papert's “ Proposal to ARPA for Research on Artificial Intelligence at MIT, 1971-1972”. p7-8. (527)

If Newell and Simon were trying to split the atom with a production systems mechanism, the AI Lab's professed philosophy fits into the “low road” route in Edward Feigenbaum's parlance. This philosophy, which encouraged wandering among diverse problems, resulted in the pursuit of diverse topics such as vision, NLR, systems architecture, and language. Less predictably, the pursuit of these discrete cognitive modalities resulted in significant theory regarding semantics and knowledge representation, specifically the idea of frames. (Frames, as we will discuss in Chapter 12, are programming structures which allocate locations to various designated semantic and syntactic aspects of a given datum. They are not the same thing as the Frame Problem). Cognitive scientific experimentation at MIT seems to have leaned more toward linguistics than toward abstract problem-solving. Other salient projects, such as Student and SHRLDU and the MicroWorlds projects, were multi-modal “one-man-bands”. Researchers in both of the latter projects swept in

both vision and natural language parsing. All of these also dealt with natural language recognition, which is not as salient in CMU or Stanford work until the ARPA-SUR research on speech understanding in the 1970s.

The Semantic Information Processing Dissertations

The AI laboratory and the more recent creations such as the Media Lab are famous in part for the work of the hackers. But the academic output of the AI community, created by people who remained on the academic track, was also and remains highly productive. During this time period, the Laboratory attracted an increasing number of students, including some who became prominent in AI and in the broader computer community. The projects undertaken include every major field in AI. We shall sketch these broadly as encompassing A) question answering; general semantic world-knowledge; and B) mathematical reasoning (528). Later on, the work of the Lab moved to include planning languages: Planner and Conniver, which also encompassed understanding natural language and the blocks world, a descendent of the geometrical scenarios of the SEE program discussed a bit earlier. We will consider the latter in the final segment of this chapter. The integrated nature of the research in the laboratory should be readily apparent in light of the fact that vision and perception were both cause and effect of the “cognitive” research.

Question Answering and General Semantic World-Knowledge

The idea of the program understanding computer is extant as early as Louis Hodes’ paper, “Some results from a pattern recognition program using LISP” (529). Hodes’ program exhibits ‘understanding’ of geometric figures in one of the simplest manners possible, as coordinates in the Cartesian plane, rather than as visual figures. The LISP lists inventory the Cartesian coordinates of triangles and squares; if queried, the program can figure out the quantity and type of triangles and other figures in the Cartesian plane which makes up its world.

A comparable program, BASEBALL, was designed and implemented at the Lincoln Labs in 1961 by Cordell Green. This program

answered questions, posed in English, about baseball games and teams, by parsing the sentence input by the user, turning the question into a canonical form, and retrieving information to answer it. The domain and form of input are both quite limited, but this was the early form of the question-answering which would be pursued in a far more sophisticated form (530).

Semantic Information Processing (1968), a volume edited by Minsky, presented the works of various of his protégés during the mid-1960s, as well as the SIR program, a dissertation by Herbert Simon's student Ross Quillian. This work included much progress in parsing pseudo-natural language and a semantics. The dissertation by Daniel Bobrow is particularly exemplary in that it neatly conducted knowledge representation in a tangible and tractable domain. As such it epitomized both the concern with linguistics and with semantics which seems to have characterized scholarship at MIT during the time period. It is difficult to avoid the inference that this focus was due in part at least to Noam Chomsky's presence in the adjacent linguistics department. Daniel Bobrow stated that Chomsky himself was not particularly fond of AI- but he was nevertheless influential (531).

Student and Semantic Networks

Daniel Bobrow's 1964 doctoral dissertation, *Student*, was one of the 'one-man bands' playing the most instruments, so to speak (532). *Student* solved simple schematic algebraic problems of the sort found in standardized college entrance examinations, although the questions are far more difficult than those in the GPS. These included word problems in which one has to figure out, for instance, the distance a vehicle will travel given a specific velocity and duration. Such problems are addressed by hierarchical layers of different modes of computation. The program initially parses the input syntactically to find algebraic operators in congruence with pre-defined pattern matching templates. It organizes a list of the items to be solved for. Invoking the next bracketed operation, the program then introduced the "solve" mode which carried out the algebraic operations. Should the problem not be embodied so simply in syntactically clear algebra, *Student* has recourse to idiosyncratic linguistic niceties of the English language, for instance,

the nature of pairs (533). The strength of work in modern linguistics has been in its ability to explain progressively more complex utterances. This has progressed from B.F. Skinner's finite state machines, to context-free grammar parsers (534). AI programs have followed this itinerary, and this was the beginning of a series of progressively wider and less formally stringent NLR programs.

AI natural language problem solvers such as this program have parsed and 'comprehended' statements in a simple subset of natural language. Such AI parsers also appear to constitute their own version of a transformational grammar because they can parse statements, and have an underlying semantics as well. Barr, Cohen and Feigenbaum's analysis of Student state that it was an intermediate step toward production systems. This, too, seems evident in the design of primitive operators with defined right- and left-hand clauses. Student follows the inductive tone of the MIT research tradition. Its regular form of inferencing clearly also indicates that Bobrow was thinking about the transformation rules as a general inferencing device.

Ross Quillian's dissertation, presenting the first version of the semantic network formalism, embodies fewer distinct modes of AI cognitive emulation. But it initiated an important form of psychological modeling (535). The semantic network formalism was an effort to embody, and be able to refer to, the meaning of words through their association with other words. Given a word, the program invokes its semantics through its placement in a web in which it is related to other words. Semantic objects, or words, are known as nodes. The relation which nodes bear to each other is designated by means of linear arcs. While we are explaining the nature of the primitives germane to Quillian's invention, the level at which a node is defined should be stated. A node is not an atom, since the latter is a token defined at the level of computer languages. Nodes in semantic nets are stated at the knowledge level, rather than at the computational level. Quillian introduced the concept of relational primitives, a construct which is recalled in the Roger Schank and Chris Riesbeck's idea of conceptual dependency, a few years later. The term specifically designates the syntactical nature of the permitted relations between nodes. To render the net manageable, only six different such types of relations between

different nodes are possible. These include relational forms which are familiar in later AI knowledge representations (536).

Quillian's program did not address episodic or recall memory. Instead it assumes that semantic memory is stored in nicely hierarchical trees, worthy of a philosophy professor, rather than a typical, chaotic human being. Above the primitive (knowledge) level, the next more general echelon of storage of words is within a plane, as it is called, which is a term for general classes of which the word is a more specific instance- that is, a meta-class (537).

Bertram Raphael also implemented his idea of semantic question answering as part of the same cohort of students as Quillian. His program would answer questions in a limited natural language. The question answer is complemented by an expanded LISP syntax which begins to flesh out sorts of relations between objects (538). The relations between objects were themselves fairly simple: A declarative sentence was interpreted as stating that a relation hold between given objects; the relationship usually concerning attributes of an object, or defining parts and subparts (p11). The simplicity of the relations embodies simply a LISP list in which objects have a property list (or description list). When the program was developed, it included only binary relations, but Raphael anticipated that unary operations- that is, adjectives- and trinary relations involving transitive verbs (p37)- were going to be added.

Raphael acknowledged the need to understand meaning in human beings' language, and made clear that the restrictions of LISP circa 1963 explained the limitations in subtlety of associations in his program. For 1963, however, this program was impressive. If this appears superficial, consider that most understanding of abstract things are embedded in relational networks of formal understanding- rather than simply being experienced phenomenology. Being in itself and the experience of the moment certainly do account for much of our qualia [felt experience], but a good deal of understanding of less warm and fuzzy things is highly formal. Ontologically relative understandings of things is, at worst, a pretty good start.

Mathematical Reasoning

A great deal of work was done in the introduction of artificial intelligence into mathematics. Mathematics is the ideal proving ground, at least for a start, for AI. More generally, of course, this had been the basis of the creation of computing itself, in the form of the rapid and accurate calculation of interminable ballistics equations. It is more restricted and austere in its syntax. It intrinsically adheres to rules of formal language rather than the foibles and oddness of human informal languages. If it is not formal, mathematics is nothing. It is limited in its semantics, and in this sense more attractive than the 'hairier' semantic fields.

'Intelligent' programs to carry out mathematics were one of the earliest forms of AI applications. This meant the creation of a program that would carry out operations itself, once instructed to do so, and later to carry out the necessary operation (e.g., simplification of an algebraic equality) with more autonomy. IBM scientist and Dartmouth conference co-sponsor Nathaniel Rochester created a "Symbol Manipulating Language" at the Artificial Intelligence Project in 1958, very early on in the game indeed (539). His program differentiated one variable of a function, and then measured the limit, in the process taking in the necessary substitutions and simplifications. At first, programs such as SAINT, the Symbolic Automatic Integrator, were relatively simple (540). But these proceeded to basically make carrying out simplifications, and the difficult 'heuristic' parts of calculus, as unnecessary as washing clothing on the riverbanks. Several years after SAINT, Joel Moses created MACSYMA, which differed from SAINT in being 'purely algorithmic' - and much faster. Where SAINT had used heuristics, this program used nine algorithms to simplify expressions, carry out substitutions, etc. MACSYMA was originally developed as one of many liaisons between MIT and a private company, in this case by an effort in symbolic mathematics by the MITRE Corporation (itself an MIT spinoff), with the support of the Air Force (541). The program taken up by Project MAC, where it had a more general effect on the development of automatic programming, and continued to be given bells and whistles and an operating environment throughout the 1960s and 1970s (542). Moses appreciated the facilitation of symbolic mathematical calculation occasioned by the development of programs like MACSYMA:

“ Considering the success of the program we believe that it is fair to predict that a program can be written in the near future which will integrate most of the problems in a standard table of integrals in a matter of seconds. When the PDP 6 computer at project MAC will have a 1 /4 million word memory, the time spent on garbage collection should be vastly reduced and chaining between the different sub-routines used by the program would not be necessary.” (543).

Where to go after this ? Two comparable programs at the end of the 1960s did, respectively, more of the same, and added natural language capacity. Late in the 1960s, the AI Lab developed MATHLAB, which was designed to be more ergonomic than previous programs, and which thus effectively made developing and solving problems in symbolic algebra easier (544). At about the same time, another AI Lab project aimed to combine natural language understanding with semantic knowledge about mathematics, and thus make the front end of a math problem solving program easier. Eugene Charniak began to replace the high school algebra capacity of STUDENT, designed by Daniel Bobrow a few years earlier, with a comparable program which solved calculus problems, also stated in a relaxed English grammar. CARPS, the Calculus rate problem solver, addressed “Loosely Stated Mathematical problems”, parsing sentences, and generating the necessary equations to address problems from geometry and calculus textbooks (545).

The rudimentary Symbol Manipulating Language, and SAINT, and MACSYMA and MATHLAB, cumulatively but increasingly turned humans doing their own calculations into an unnecessary craft. It meant that the teaching of calculus to students, and making them solving integral and differential equations themselves, was as unnecessary as the teaching of long division to those recent generations of middle school students who have carried calculators. Lest we forget, let's note that a massive human labor that used to be involved in such things; it is not odd that this was one of the first tasks which AI took up and conquered. Theoretically, computers at this time could indeed carry out such functions. But it was not as if the service was easily obtained. Instead, a problem had to be considered 'important' and specially submitted to the machine operators, tortuously programmed in a machine or assembly language. This application brought quicker solutions to math problems much more proximate to more people- even if not to

many more people. We can say that symbolic mathematics, or calculus, was the first domain in which AI was triumphant.

System and Institutional Developments

Even people entirely uninterested in housekeeping need to take out the rubbish and do the dishes sometimes- or at least delegate such necessities. These matters, in several different ways of speaking, were indeed cared for at the MIT AI Lab. Some of it was indeed, delegating. But at least through the 1960s, Minsky and Papert wrote reports- excellent, thoughtful reports- in a style that clearly did not suit the deadline and military applications style of the new, budget-conscious ARPA.

The MicroWorlds dissertations clustered around the end of the 1960s, just as the Project MAC group split off into the AI Laboratory. Just two years later, Minsky and Papert both relinquished their formal leadership of the Lab, not willing to spend so much time in the chilly military-tinged research administration climate. They were fatigued with confrontations with the 1970 passage of the Mansfield Amendment, which formalized and restricted military-supported research as discussed in Chapter 8. Patrick Winston was designated Director of the Laboratory in 1972, an event that he himself refers to with irreverence:

“...the date is a little obscure because I became acting director and was acting director for an undetermined period of time, and then I became actual director, as Marvin says.” (546).

Meanwhile, the AI Laboratory endured in absolute numbers, but shrank in relative prominence as a bastion of computer science work at MIT. The 1968-1969 AI Lab Report lists roughly sixty people at the AI Lab, which was still technically, part of Project MAC. The AI Lab became independent of Project MAC in bureaucratic terms near the end of 1970, but did not balloon in size, although its parent entity was much larger (547). In 1971, there were more than three hundred people working at Project MAC (548).

Moreover, there was plenty of ongoing research in AI at BBN, Lincoln Laboratories, the RLE, and only a few years later, the Media

Laboratory, which would begin in the winter of 1985-1986 (549). The LCS, established several years later, also carried out systems work. This lab however was concerned with the essence of computer systems rather than the nature of intelligence.

A tremendous portion of the work of both the AI Lab and much other work at MIT was carried out as systems work, as well as in the form of the discrete AI programs discussed elsewhere in this chapter. Project MAC, the mother of all time-sharing systems, was initiated in 1962 (as we saw in Chapter Four). It has basically continued since then- perhaps this definition could set the effective date of the development of the ARPAnet another seven or eight years earlier. Repeated iterations of Project MAC allowed the development of concurrency, remote usage, relatively rapid access to programs stored in core memory, and instigated as well dramatic improvements in user protocols. Subsequent iterations under different names, such as MULTICS and CTSS, permitted and facilitated the AI laboratory work (550). As the work progressed, new issues such as hackers and password protection were brought to the fore. Levy's history of hacking- in which the hackers are always the heroes- chronicles the earliest introduction of the morass of password issues, in which a utopian socialist plot (for lack of a better word) involving repeated destruction of password protection schemes threatened the Lab's participation in the ARPAnet. This is not a popular perspective nowadays- people who want to sell information are not in a good position to claim that "information wants to be free".

MicroWorlds and Planning Languages

Good questions may be too good- for instance, global existential ones such as "What is the meaning of life ?" tend to make college freshmen depressed (or used to make college freshmen depressed). 'What is intelligence, and how do we embody it ?' may be seen as such a question, which needs be handled properly lest it destroy itself in metaphysics. The papers produced by the Lab in the late 1960s state that the Lab's interest was in the larger whole of what intelligence was rather than in improved formal systems (e.g., automatization of deduction, or theorem proving). This meant effectively engaging problems which presented large clusters

projects, which forced the development of numerous sensory modalities and had numerous attached challenges:

“ The long run goals are concerned with simplifying, unifying and extending the techniques of heuristic programming....

We believe the most important problems in the field of AI are centered, today, around problems of using many different kinds of knowledge in the same system. We think that working on such problems is the most direct path toward finding out how to build systems with greater “generality”- systems that do not need to be rebuilt whenever the problems have to solve are changed slightly.” (551)

MicroWorlds

Even if good questions can be too good, they need not be completely overwhelming. At this particular institution, good questions which were too good were handled very well, in the form of a large cluster of problems called MicroWorlds. MicroWorlds was a series of experiments within MIT’s Project MAC, cultivated in the Lab as a testbed for the sundry interests in vision, robotics, natural language, and knowledge representation of semantics. The projects took SEE, the vision project that had made so clear that semantics and world knowledge were intertwined with vision, to a higher level involving robotics and natural language- ‘See squared’, we might say. The goal of their inquiry was manipulating objects in a toy world in these various modes.

“...Each model or “microworld” as we shall call it- is very schematic; it talks about a fairyland in which things are so simplified that almost every statement about them would be literally false if asserted about the real world.

“ Nevertheless we feel that the [MicroWorlds] are so important that we are assigning a large portion of our effort towards developing a collection of these micro-worlds and finding how to use the suggestive and predictive powers of the models without being overcome by their incompatibility with the literal truth.” (552)

The Microworld consisted of a number of square, rectangular and pyramidal building blocks of the type used by children, in several colors, and boxes in which to place those blocks. These blocks were stacked and placed in front of, behind, and next to each other. The world is three-dimensional. The AI programs which Minsky and Papert hoped to procure were capable of simulating intelligence by

observing the movements of the blocks features and placement (vision, NLU, semantics, KR), or of effecting the movements themselves (robotics, vision), according to (constrained) natural language input. The program was also intended to be able to answer questions about the blocks.

The MicroWorlds problematique spawned a cluster of important dissertations during the turn of the 1970s. The graduate students who created these programs was one of the notable cohorts of AI's history, including Gerald Sussman, Terry Winograd, Eugene Charniak, John McDermott, David Waltz, and Patrick Winston. All were successful, often in orthogonal categories and different representational modalities, despite their common quality of all being very good.

The MicroWorlds opus inspired two other less immediately obvious developments. These are the frame formalism for knowledge representation, which we will discuss in Segment Nine of this chapter, and planning languages and programs, which we will discuss momentarily. The apparent inspiration for frames was the vision research conducted in the MicroWorlds period (Minsky 1975). The other primary development, planning languages and programs, emanated beyond the MIT milieu.

SHRDLU

One of the most successful of the MicroWorlds cohort was Terry Winograd's 1972 doctoral dissertation research project, which resulted in a computer program named SHRDLU (553). The whimsical name appears to have been a ruse which Winograd perpetrated by telling people that these were the seventh through twelfth most frequently used consonants in the English language. Of course, 'U' is not a consonant, although this is only a minor glitch relative to the ruse itself. SHRLDU is actually a name for mythical monsters featured in MAD magazine (554).

This program was a toy universe made up of blocks and pyramids in several primary colors, on a table top. The program's raison d'être was the manipulation of these virtual blocks, at the user's request. This repertoire of manipulating these two-dimensional shapes included the program's evaluation of the states of the (Blocks)

World in various ways. The user could request that the program, through the agent of a robotic arm, count the total number of blocks in the World, pick up a block of a certain color, and stack up blocks, as well as to evaluate and refuse to perform illegal moves such as stacking unstackable blocks, for instance, blocks on top of pyramids. Unlike certain other MicroWorlds experiments, this Blocks world existed only in computational form. The program moved these blocks using a simulated robotic arm. There were no material sensors and effectors, except the inputs provided computationally within the program itself. The virtual Blocks world was projected onto a TV screen in line drawings, the user-program dialogue displayed under the drawings (555).

SHRDLU worked by parsing the NL commands typed in simple declarative and imperative sentences by the user, restating them in a semantic format meaningful to its KB (the parser and the program for semantic analysis were separate components), and responding by performing the action requested (the problem solver was still another component, although a simple one). Its toy world was literally that- the simplest of rudiments and largely declarative. In contrast, the GPS world necessarily entailed several levels of search; SHRDLU leaned toward knowledge representation rather than problem solving. In order to map the input into the knowledge base, the program did not need to engage in any sort of search. This stage was entirely circumvented by the immediate association of every declarative object with related procedural activities. In SHRDLU, predicates and noun phrases were entirely integrated. Attributes and relational functions were directly mapped onto the semantics of the world. This means that the concept of 'on top of', an important imperative for the user, was directly related to those objects which could be placed on top of other objects. Illegal moves were precluded by the structure of the language itself. Winograd's stated intention was to illustrate the close intertwining of these in real-time activity. Because of the direct contiguity of semantic objects and attached predicates, search space was curtailed- or rather, decapitated. The program succeeded in demonstrating the practicability of this semantic and syntactic intertwining, of the immediate compilation of predicates with objects- for its stated and limited toy realm. The program in this way entirely sidestepped the issue of the analytical search engine so central to other AI

subcultures. SHRDLU was written in LISP and MicroPlanner, and this was one of the keys to its success. As we will see momentarily, planning languages, of which this was the first, worked as ongoing discrimination net programs. They consisted of lists of instructions, from which [predicate or object] elements could be added or deleted, or compiled into macro-operators (556).

This design meant that SHRDLU “knew” what it “knew” only within the Blocks world it inhabited. It was rather a ‘Blocksheaded’ program, despite the fact that it actually worked. There was no meta-knowledge. Given an order or request to manipulate or evaluate a particular object, it responded because of its knowledge about that individual. But it could not generalize: it was told what was possible and impossible only because of direct mappings, not because of any abstract knowledge about the integrity of solids (solid objects can collide but not coexist in the same space), or the nature of geometric shapes. SHRDLU had minimal knowledge about physical objects. It knew this very directly, as illegal and legal movements of the blocks and pyramids were precisely ordered. While this is probably the most famous program to have come from the MicroWorlds years, it is interesting to see that other programs actually did succeed in the development of meta-knowledge. Patrick Winston’s Arch-Learner program, in contrast, actually conducted induction. It identified arches in visual settings. SHRDLU could not really be generalized, because these predicates all applied as arguments to specific semantic objects, not to meta-objects of any sort. It could not be applied outside of the Blocks universe, because there was no generalization of knowledge. While this modus operandi was an efficient planner, and shared this trait with other planning programs, it did not carry larger implications of cognitive verisimilitude. It seems that SHRDLU paralleled later expert systems in the much-criticized narrowness and brittleness of both systems. Knowledge based or expert systems of course, were mostly diagnostic rather than planning, and encompassed often sizable volumes of information. (But like this MicroWorlds project, they were actually born from a low-road experiment in applied AI). Any honest commentary must contemplate this program both as a successful hack- and more than a hack- as well as an artifact that only treaded water in terms of knowledge representation. This program both reached the limits of specific embodiment of toy

worlds, and illustrated the limitations of programs lacking meta-knowledge. SHRDLU's performance success is precisely offset by its location at a methodological dead end. SHRDLU is entirely concerned with its own blocks microworld and not at all with the universal nature of geometric shapes, how such shapes might balance on top of each other (and the common-sense physics implied therein), or similar essentials of commonsense physics. This is an extreme example of both the benefits and the limitations of a closed world. Most importantly, it worked: it was successful in its demonstrations and it did what it claimed it could do.

PLANNER and CONNIVER

The MicroWorlds initiative instigated another significant development, that of planning languages. This is just as important as the semantic improvements of the MicroWorlds ensemble. Carl Hewitt, who wrote PLANNER, the first planning language, was an MIT staff researcher. Planner was based on LISP, and seems to be in local design form a superannuated LISP list structure. Programs consisted of lists of declarative functions and variable information. In addition to imperative programs which included semantic data and procedures concerning objects, the language included overall theorems concerning courses of action; hence the reference to a planning language. This embodied control and representational knowledge; the instructions were changeable simply by adding or deleting a given instruction line. Planner could evaluate new data through its procedural theorem component. It could also render a declarative statement such as A therefore B, as an imperative one (557). Planner also implemented demons, borrowing the term from Oliver Selfridge, which acted as global and persistent agents to implement certain procedure calls.

The Planner language project was undertaken from the orientation of developing specialized LISP-based languages for planning. It also included many features of an entire operating environment, including demons which acted as persistent objects for any number of applications programs (chess, vision and robotics); and protocols for building a semantics and knowledge base—"declaratives, imperatives, goals and deductions" (558).

This project worked extremely well, and was wildly popular as a practical, rapid-deployment edifice, and was used in several of the MicroWorlds oeuvres (in SHRDLU; in Gerald Sussman's HACKER, a model of the process of acquiring programming skills; and in Eugene Charniak's NL comprehension model, STORIES; Minsky and Papert," 1968-1969 Progress Report." MIT AI Memo 200). Winograd, Sussman, Charniak and Greenblatt together built MICRO PLANNER, a simplified form of the language (Minsky and Papert," Proposal to ARPA for Research on Artificial Intelligence 1970-1971." MIT AI Memo 185. December 1970, p11). Planner was followed by many comparable planning languages similar in structure (Stanford A.. I.. Language, or SAIL, POP-2, and QA4).

Conclusion: The LISP Machine and Industrial Applications in the 1970s

Both as a result of the intrinsic tropisms at the AI Lab, and ostensibly because of outside pressure, the Laboratory began to consider developing its own hardware and industrial applications late in the 1960s. The idea of MIT-produced and mass-distributed hardware proved to be unsuccessful, for reasons that are more apparent in retrospect than they were at the time. Why should a university laboratory needed to develop its own hardware ? Wondering why they were inspired to do so is perhaps inevitable for anyone thinking about the topic after 1990: why bother with the immense work involved, when a commercial computer maker will gladly produce one to your precise specifications ? However, the environment was different at that time, and besides the "small" and "cheap" scientific computers made by DEC, there were fewer options. Minsky and a number of others at the laboratory, including Stallman, also believed, politically and ideologically, in the wider distribution of computing machinery (559) .

ARPA discouraged this movement, for several reasons. The ILLIAC IV, a hardware effort undertaken by ARPA, had been a mixed success. ARPA was also equivocal about supporting computer development with the extremely cheap labor of graduate students, rather than the full-time pay of engineers- the same people, a few years later in their lives (560). The people engaged in 'growing their own' did not care much what ARPA thought (561). The homegrown LISP machine movement proved ultimately abortive.

MIT's two hardware spinoff companies, LMI and Symbolics, were not exceptionally commercially successful. The two warring companies which spun off from MIT's Labs, did not do well. AI labs did much better when they stuck to software and to specific architecture modifications, as in the c.mmp parallel computing projects at CMU and the Connection machine at MIT, rather than the entire box and the hoopla surrounding it. However, the AI Lab's Lisp Machine was, during the early 1980s prior to its entire eclipse by Sun and Apollo and other workstations, specially and usefully devoted to running Lisp and including an assembler and a specialized microprocessor (562).

Serious and successful attempts to actually construct working computer hardware (the entire box, that is), had to await the entry into the (initially nonexistent) market of the serious commercial contenders, a decade later.

A more constructive note is struck by the move to the prospective industrial applications of AI, which started under Minsky and Papert around 1970. The Laboratory had begun to entice ARPA with the bait of interest in things that were highly useful for military and industrial purposes. The Mini-Robots project, circa 1972, was of a practical, even mundane tone. A miniature hand-eye system- i.e., a "mini-robot", could be bent to many industrial applications:

" 2.1 general production

- Application of automation to assembly, testing, and finishing; more responsive robot assembly, mass customization- does not use this term but this is what they mean;

...

2.3 mining and undersea resources;

2.4 housing- prefabrication to cheapen housing;

2.5 transportation; sensing devices to improve moving belts and other [urban ?] people moving things;

2.6 space; centralized manipulator-sensor systems to operate all onboard systems... "

(563).

The predisposition toward practicality continued, and grew, throughout the 1970s. The MIT AI Lab's 1976 ARPA Proposal, with Pat Winston's name on it, did have a different and even more

serious tone to it than did the more theoretical and intellectually diffuse papers in the 1960s (for instance, Minsky and Papert's 1968-1969 ARPA statement). Winston was very obviously concerned with showing how AI could lead to industrially productive technology, and with suggesting end-uses for AI.

" The application of the ideas of AI to computer vision and manipulation for productivity technology, vehicle guidance and image understanding... are only what we view as immediate areas of maximum opportunity. Many others are suggested by the following sample:

SOME AREAS OF APPLICATION

Manufacturing

- assembling electronic systems

- assembling small mechanical devices

mining

- coal mining in dangerous mines

- undersea recovery of manganese nodules;

farming

- selective harvesting with mixed crops

- pest control

- pruning

repair and maintenance

- vehicle debugging

- floor care

purposeful monitoring

- intelligent autopilots

- shipboard functions

automatic programming

- management reports

- inventory control

- production scheduling

- quality control;

management assistance

- very large databases

- intelligent PERT networks

- scheduling people and groups

- question referral

- news summarizing

- document draft polishing

logistics

- routing

- intelligent substitution

equipmentssharing
.... (564).

The list, all of it practicable and thoughtful, goes on for another page.

Chapter 12. Developments in Hardware, Software, and Utilities in the 1970s

Introduction

AI's founders and leaders were protagonists in this story, and astonishingly prescient. But Minsky, McCarthy, Newell and Simon and their growing circle were neither omniscient nor omnipotent. Like Turing, they could envision programs that could not yet be carried out by the hardware of their time. Otherwise, they could only do what the computers could handle, and for that the larger field of computing needed hardware engineers.

Like the proverbial rising tide that lifts all boats, the waves created by the microprocessor had broad, comprehensive ripples in every aspect of computing. The minute the microprocessor was invented, the personal computer was not far behind. Indeed, the PC was implicit in the microprocessor, and this would eventually mean possibilities that had not yet seemed possible at its incarnation.

IBM began to lose its technical lead and vitality during the 1960s, while maintaining the reliability and the gravitas of its research facilities. Certainly DEC replaced IBM in the role of major supporter of scientific computing, or at least computing for AI. This role is important to our story, even to its highly intellectual components, because power and speed and memory were necessary for research computing. The major companies were the grand actors in increasing production, implementing cheaper and faster logic and memory and storage, manufacturing Flexowriters and consoles to read at them, and dropping costs, and thus all the while turning the trickle of output of computers into a stream and then a flood. Through the course of the 1960s, visible change in implementation of the computers through businesses and government functions took place as the supply increased immensely- more precisely, ten-

fold relative to 1950 (565). The manufacturers became bigger companies: IBM moved into the Fortune 10 during the 1960s, and gathered more than 50% of the market in computing itself (566).

In addition to the general effect of increased speed and robustness in processing and storage through the IC, distinct systems improvements were initiated during the 1960s, and carried through to completely alter the information technology world- and far beyond “just” computing as well. The role of AI and its larger research environment is both enormous and inadequately appreciated. These changes, in the form of Unix C C++ CP/M and MS-Dos; and the prototype of the personal computer, came from research labs supported by IPTO in the 1960s. The result of these innovations included scientific computing in the form of workstations for scientific and technical computing (Sun Microsystems most notably); and the personal computer running MS-DOS, which was ultimately a derivative of UNIX, Multics, and the Project Mac research of the early 1960s.

Alternatively, the aesthetic design, input-output innovations, networking innovations, and desktop, largely came from SRI's Augmentation Research Center, Sutherland and Kaye at the University of Utah, and Xerox PARC- as Allen Newell said, BBN West. The commercial realization of the research instigated, funded, and encouraged by the IPTO in the 1960s, thus was in itself enormous apart from the knowledge engineering. engineered artifacts, and other efforts at embodiment as achieved by AI during this time period.

The MicroProcessor and its Role in Developments through the 1970s

IBM has come under wide criticism in the computer histories for being slow to implement timesharing. The hackers and programmers may rail, but at least at first, Big Blue was observing the reality of circumstances. Hardware ruled, and software accommodated its restrictions. At least, hardware ruled until the integrated circuit was widely implemented, which only pertained by the early 1970s. The big computer manufacturers may be faulted for not energetically implementing timesharing and superior input-output for the scientific and hobbyist markets late in the 1960s, but

in truth they did encase new IC arithmetic and logic units behind steel cabinets and market them quite rapidly. Sometimes even too rapidly, as in the case of the IBM 360, which introduced a close relative of IC logic, as well as upward migration paths and massive system improvements in the mid-1960s and initially lost money.

As we mentioned earlier, DEC appears increasingly prominent throughout the 1960s,, but remained minor in the larger role of computing companies. The major companies seem to become somewhat faceless, but their role in the sweeping introduction of access to computing into the university does not diminish. This is simply because they were the grand actors in increasing production, implementing cheaper and faster logic and memory and storage, manufacturing Flexowriters and consoles to read at them, and dropping costs, and thus all the while turning the trickle of output of computers into a stream and then a flood. Through the course of the 1960s, visible change in implementation of the computers through businesses and government functions took place as the supply increased immensely.

The other protagonist in the sheer juggernaut of faster computing was a technological force, more specifically the torrential flood of integrated circuits being produced and let loose on the world in more and more computers. The application of integrated miniature circuitry to logic, and then to memory, was succeeded by the miniaturization of the entirety of computer functions in one unit at the turn of the 1970s. We saw in Chapter Three that Fairchild Semiconductor, the first of many firms founded by the defectors from William Shockley's shop, had become very successful at applying the integration of circuitry to logic. Robert Noyce, Gordon Moore, and Andrew Grove moved left Fairchild to found Intel in 1969, in order to concentrate on custom-designed integrated circuits for specific products (567).

As Noyce explains,

"...the cost per bit (binary digit) of random-access memory has declined an average of 35 percent per year since 1970, when the major growth in the adoption of semiconductor memory elements got under way. These cost declines were accomplished not only by

the traditional learning process but also by the integration of more bits into each integrated circuit: in 1970 a change was made from 256 bits to 1,024 bits per circuit and now the number of bits is in the process of jumping from 4,096 per circuit to 16,384.”

...After the introduction of the integrated circuit in the early 1960's the total world consumption of integrated circuits rose rapidly, reaching a value of nearly \$1 billion in 1970. By 1976 world consumption had more than tripled, to \$3.5 billion. Of this total U.S.-based companies produced more than \$2.5 billion, or some 70 percent, about \$1 billion of which was exported to foreign customers.” (568).

Nor was there an end in sight. In 1964 Gordon Moore, then director of research at Fairchild Semiconductor, saw that the number of elements (roughly speaking, the complexity) in integrated circuits had been doubling every year since their invention in 1959. The scale of miniaturization continued to grow, or rather shrink (569). According to Moore’s Law, this trend will go on, and for the purpose of studying 20th-century AI, this law had no loopholes.

Intel's particular angle on ICs was the introduction of miniaturization to data storage. In 1971, Intel engineer Ted Hoff, at the time still a Stanford graduate student, began to design an IC for a scientific calculator for Japanese calculator manufacturer Busicom. The 4004, as the 1971 chip designed for this calculator was known, was programmable rather than being specifically designated to certain functions. This versatility allowed its logic functions to be complemented by other logic units to make up a good portion of the functions of a computer. Called the ‘computer on a chip’, or more formally, the microprocessor, it set off a new era in miniaturization and dropping cost of processing (570).

As the aphorism tells us, nature abhors a vacuum. This appears to be the case, in the instance of old technologies. Internal memory was still implemented in the form of magnetic cores. The new chips which Intel introduced in its first several years were called memory chips. This product was immediately in fierce demand, and the unit price of computer memory (for the 1103 memory chip) dropped as production skyrocketed (571). Along with Intel, several dozen firms, all with silly names and initially composed of defectors from

Fairchild Semiconductor, and all, appeared. The fantastic flood of hardware assured computer users of no end of opportunity. Hardware developments subsequent to the IC included DEC's introduction of further versions of the PDP. Research-oriented computing in general would be greatly improved as DEC itself split into two, and the succeeding company, Data General, was matched by a number of other minicomputer producers.

Applications and Software for PC computing

Superior microprocessor capacity- and the existence of the microprocessor itself, was what made possible personal computers and workstations. In turn, both of which allowed a far greater commercial and consumer market than anyone had imagined could exist. The mass commercialization of computers, eventually for both personal use (e.g., consumer chat rooms on casual topics), and as small capital goods, for instance for tiny businesses, in a sense started with the integrated circuit. This development toward computing in every home and office proceeded gradually, and by the early 1980s its progress allowed AI to become oriented to industrial needs as well as to scientific issues.

This core technical move forward meant that there were more extensive possibilities for products than had previously existed. The IC and the microprocessor led to the era of (relatively) cheap microcomputers for smaller businesses during the 1970s, but something ultimately more substantively important for society was now a possibility. Aspray and Campbell-Kelly tell us that:

“ It would have been technically possible to produce an affordable personal computer (costing less than \$2000 say) anytime after the launch of the 4004, in November 1971.”(572).

Yet the entire combination of ancillary and complementary technologies took considerably longer to assemble. The first Apple appeared more than five years after 1971. The problem was one of application or development, rather than of research. There was for a long time, no vision on the supply side (companies producing a product), nor a perceived market on the demand side. Granted, there would have been some technical hurdles: logic, memory and

storage were in the form of very large chips, storage was often on delicate tape drives or on very bulky and even more delicate floppy drives, and punch cards were still in use. The display monitor would need considerable improvement, a greatly increased buffer zone allocated to it, and a great deal of software for 'personal' computing had as yet to be developed. Finally, the size of all of the components would have had to be much smaller before anyone but a hobbyist would have wanted such an item.

This was all of the 'development' rather than the 'research' side of R&D, however. But the average hobbyist did not have any vision of a personal computer as we know it, and Douglas Engelbart and a few of his colleagues were not in the business of putting out commercial products. Neither was the Xerox Corporation, which failed very publicly with its too-early efforts to produce a commercial Alto. Thus the commercial demand for this product was not present immediately. Personal computing had to start as a movement of hobbyists, and its early incarnation was simply a low-budget version of existing mainframe computing, complete with teletype machines and paper tape punches and readers.

The first "personal computer", the Altair, has earned its place in history by virtue of sheer audacity rather than technical delicacy. The machine did not even have a display or a keyboard: one communicated with it by flipping the hand switches in its front, using pure binary language. In addition, the Altair was not even sold as a finished product but as a kit (573). I/O devices, printers, keyboards, and external storage devices were sorely lacking. Very soon after the this primitive prototype appeared, Bill Gates (1955-) and Paul Allen (1955-) wrote a Basic language program for the Altair. They formed a company to sell it, and then began vigorously defending their property rights to the language. Bill Gates' wrote his famous 'Open Letter to Hobbyists', defending programmers' right to get paid (574). Steve Wozniak built a computer around the Mostek 6502 chip, and several competing small computer companies arose, and hobbyists who had spent their time building blue boxes (for telephone hacking) turned to computers (575). The rest is history.

While non-institutional computer users had existed even before the microprocessor did (576), the enthusiastic community that arose

surrounding self-built computers was thickest in the Bay Area. AI's founders were generally software people. However, all of the founders were generally strongly inclined toward accessibility and the generalization of access to computing. The first meeting of the infamous Homebrew club, influential to Steve Wozniak, was hosted by John McCarthy.

The new world of PC computing did not immediately offer sophisticated software opportunities for AI; for that, need to look at scientific computing which became far more economical during the 1970s, and which we will address later in this chapter.

The Contributions of the IPTO

No one has refuted the fundamental brilliance of the early PC computing movement. Still, many of the best technical features of the PC did not emerge from the hobbyist community or from its counter-cultural wing, but from the IPTO. Much of what was first sold as the Apple computer's Lisa machine in 1982 was, in many features, an Alto. The bit-mapped screen, the presentation of the CRT screen as a desktop rather than just a chunk of text, windows, icons, the computer mouse, and the graphical user interface, have now become nearly universalized (577).

Most of these were developed through the auspices of ARPA-IPTO through the University of Utah, SRI, and Xerox PARC. Specifically, they were developed at SRI with money from the IPTO and transported to PARC when most of Douglas Engelbart's crew moved. Both Kaye and Engelbart conducted work which was directly intermediate to the personal computer. Engelbart had been thinking of electronic offices since the mid-1950s- as we saw in Chapter 7, he had tried to study computer science at Stanford, and been rebuffed, before the field existed. Much later, following the development of the mouse under the ARC at SRI, he had demonstrated the use of the mouse at an ARPA Principal Investigators' conference in 1966, and at the famed AFIP conference in 1968 (578).

In 1974, Engelbart and his colleagues at ARC did some experimentation with a computer and Augment/ NLS system based

on the 4004 chip (579). Some of these features had also been developed at the research laboratory at the University of Utah, managed by longtime IPTO grant recipient David Evans. Alan Kaye had attended graduate school there prior to moving to California to work at Xerox PARC. Alan Kaye had written about many of these ideas in his 1967 Ph.D thesis, calling the hypothetical computing machine the Reactive Engine.

At Xerox Parc, Alan Kay and Dan Ingalls wrote the first version of Smalltalk in 1972 (580). Smalltalk was the major development language for the Alto, and its roots were strongly associated with AI. Its direct ancestor was SIMULA, which had been created for the Dynabook, Kay's hypothetical handheld computer. (The first version written in BASIC, then later versions written in Nova assembly language). Smalltalk '74 thus offered the first implemented full multi-window interface with bit-mapped graphics, an input-output convention which we have become accustomed to but which was revolutionary at the time (581).

In the early 1970s, Xerox PARC- which Newell referred to as BBN West- developed a comprehensive prototype of the personal computer. The Alto, connected to the earliest Ethernet with LAN and network capacities, included a bit-mapped high-resolution screen (an update of Engelbart's innovation), and external memory storage (64K, which was soon increased to 256K (582).

Delays in the Implementation of the PC

The PC, oddly, appeared before there was even the most trivial software to help support it. Showing up early to the party did not discourage its enthusiastic fan club, who were already working on hobbyist projects. Hobbyist projects and those sponsored by ARPA through think tanks and universities seem to have been the only viable options in personal computing in the strict sense for several years. In fact, there was no benefit at all to more ergonomic early arrivals. IBM had introduced a desktop computer for the scientific market in 1975. The model 5100 did not sell well, however (583). The Star System, a word processing and file storage environment introduced by Xerox PARC in the late 1970s, also did very poorly. This can only indicate that the very first flurry of excitement about

the Apple PC was based on hobbyist use, which was impervious to the lack of applications and the rotten display and storage facilities. The early word processing programs, such as Magic Windows (which RES used in 1981), were uniformly terrible. And the environment itself was worse. Until the Macintosh, with numerous features resembling the Alto, was introduced in 1984, personal computers communicated with the user through a disk operating system (DOS).

The most popular, by default rather than due to its innate appeal, was MS-DOS, the Microsoft disk operating system. The key to its dubious charm was its close resemblance to UNIX:

“ Like so much early in personal computing, DOS was derived from mainframe technology- in this case the notoriously efficient but intimidating Unix operating system. The early DOS style operating systems were little better than those on the mainframes and minicomputers from which they were descended.” (584).

The DOS interface was often incomprehensible to people who lacked a computer background, and terribly homely to those who had one. It had an unyielding command line interface and commands that lacked mnemonic appeal. The introduction of the GUI or WIMP interface (windows icons mouse pull-down menus), was a giant innovation that helped increase the non-hobbyist market a great deal. So was the entry of the established and paternal IBM into the zany and offbeat PC world.

But the I/O and peripherals began to appear rather rapidly throughout the early 1980s, as supply and demand began to synchronize. Finally, such things started to make computing seem comfortable and somewhat more immediately gratifying than punching paper tape or IBM cards, or using delicate and unreliable giant diskettes. Despite the wild disparity in coordination of the diverse technologies which went into the PC, the product was and is one of the most successful concepts in history:

“ In 1981, the first year for which statistics are available, the total market for personal computer software was \$140 million. This was to grow more than tenfold, to \$1.6 billion, in the next three years.

Despite the large number of entrants in the industry, a small number of products quickly emerged as market leaders. These included the Visicalc spreadsheet, the Wordstar word processor, and the dBase database. By the end of 1983, these three products dominated their respective markets with cumulative sales of 800,000, 700,000 and 150,000 respectively.” (585).

Minicomputers, Unix and scientific computing; The Improvement of Scientific Machines, 1965 through 1980

The personal computer receives the glamor award in studying computing hardware in the 1970s, largely because it has become a consumer device rather than one solely used in industry. However, we will briefly consider the lineage initiated by timesharing with project Mac in the early 1960s, through the larger projects (Multics and CTSS), Unix and the subsequent emergence of C and C++. This, of course, was ancestral to Java, the most widely used computer language for web applications in a later time period.

In 1969, the term minicomputer came into parlance, and small computers increasingly became a major sector of the computer industry; DEC, Data General (a DEC spinoff); Prime Computer and Scientific Data Systems, all became major international firms. Larger firms such as Hewlett-Packard, Harris, and Honeywell all also made small computers (586). These were almost always too expensive for individual users: the CDC 160, for instance, cost more than \$100,000. But this still meant more terminals, more users, and wider access. It would take IBM more than a decade to begin producing microcomputers- even smaller Intel-based and open architecture machines (587). However, it appears that once minicomputers began to be produced, the tropism toward workstations- culminating in the Sun workstation a decade later- would slowly accelerate. Along with this would come the cultivation of Unix and important derivatives such as C++ (588). A direct descendant of Project MAC and its numerous complementary projects, this would allow the secularization- so to speak- of programming languages invested with the virtues of Lisp, outside the inevitably smaller world of AI.

Chapter 13. Big Ideas and Production Systems in the 1970s

Introduction: The State of the Art Circa the Early 1970s

“ There weren't any sudden transitions until about '72 or '73. ...at the end of the '50s the problem was [that] we knew how to solve some problems like proving theorems by making a big search and we needed heuristics to make the search smaller. And then, at least in my side of the thing, we began to sense that heuristics would not go very far really when you had a big tree. So, the emphasis shifted to saying, in order to solve a problem the machine has to know, has to have built-in knowledge of what's relative to that field, so we gradually turned to finding new ways to represent knowledge and building up these knowledge bases and that led to these kind of expert systems.

NORBERG: And you said that transition is in the early '70s?

MINSKY: Well the transition of saying we've got to get the machine to have the knowledge for doing these things, that's in the middle sixties.” (589)

In this 1989 interview, Minsky observes a tipping point at which it became necessary to focus on representations of knowledge rather than solely heuristics. By the early to mid-1970s, the field of Artificial Intelligence had developed into a robust area of both scientific inquiry and more early applications than one might think. GOFAL continued in the form of the Dendral Project and early work in naive physics at Stanford, production systems and a great deal of applied cognitive science at Carnegie-Mellon, and a number of new projects in knowledge representation at several sites.

We will discuss these highly influential and synthetic ideas- Scripts, frames and FRL, KRL, and semantic representations- in segments Two and Three, respectively. Even in the early 1970s, there were more instances of applied AI than one might imagine that there might be. These included things such as computer vision, speech processing, and robotic implementation (590). The studies of embodiment drew from as well as led back into AI as a knowledge-related area. Moreover, places where this research was carried out,

such as MIT and the Stanford AI Lab, were also often very active in search, memory and KR as well.

Perhaps because of the increasing number of applications in commercial and military areas, there were even more continued efforts to link database facilities to natural-language query programs. For instance, BBN undertook at least two projects of this sort early in the 1970s (591). Moreover, the affiliated projects at MIT, such as SHRDLU and Charniak's work, emphasized the natural language processing needed for real AI.

As part of any 'state of the art' statement about AI in the early and mid-1970s, we must take into account the effect of the early and dramatic success of the ARPAnet. Frank Heart, a BBN scientist and administrator, observed at an ARPA PI meeting in 1973:

" During the past year, the network has nearly doubled in number of nodes and has experienced more than an order of magnitude increase in daily traffic."(592)

The growth in traffic fed a need for applications. These proliferated, including, for instance, software that facilitated automatic routing of messages, technical staff on duty every single hour of every single day at ARPAnet's national coordination center, the NCC (593), the simple minded file system or SMFS, for short-term storage of files while using the net (done at UCSB), early distributed file systems developed at UCSB, MITRE, and SRI, and the first networking over a satellite link (50 kb, to Hawaii) (594). The ARPAnet served several purposes, greatly helping AI practitioners to communicate, and giving them an opportunity to start to develop automatic programming applications which would be practical and widely used. Even as early as the 1971 ARPA-IPTO Principal Investigators' meeting, the increasing practicality of AI was clear.

The state of the art in AI is also well-illustrated by the sheer fact of the increasing number of institutions which participated in meetings and conferences. At the 1971 ARPA IPTO meeting, the list was much more extensive than the initial list of those who J.C.R. Licklider picked out (595). The predictable inventory of CMU, MIT, SRI and Stanford, includes as well UCB, BBN, Harvard, Washington

University, the University of Utah, Network Analysis Corporation, Case Western Reserve University, UCSB, Applied Data Research, UCLA, Dartmouth, and the U of Illinois. Other institutions include BBN, Rand, the System Development Corporation, and Lincoln Labs. Nor is this list entirely exhaustive: it was at about this time that Xerox PARC and ISI at the University of Southern California were established.

Convergent Concepts in the Early 1970s

“ It seems to me that the ingredients of most theories in artificial intelligence and in psychology have been on the whole too minute, local, and unstructured to account- either practically or phenomenologically- for the effectiveness of common-sense thought. The chunks of reasoning, memory and perception ought to be larger and more structured, their factual and procedural contents must be more intimately connected in order to explain the apparent power and speed of mental activities.” (596)

Minsky's insight, published in 1975 in " A framework for representing knowledge", was galvanizing, as always, but spoke of a larger zeitgeist as well. The General Problem Solver and kindred projects in AI's first fifteen years were composed of small spaces of semantic knowledge, in which no one maintained more than a passing interest. This lack of intrinsic interest in the semantics of Monkeys and Bananas and checkers was intentional (sic). Hardware and the establishment of taxonomy of sorts of search overrode the interest in the content of the problem being examined or the knowledge being represented. In contrast, attention was focused upon search procedures. This had started to change by the end of the 1960s, and by the middle part of the 1970s was rather entrenched. The sheer intellectual weight devoted to Dendral, and related projects such as Meta-Dendral and the later medical programs, helped to practically reverse the proportional weights of knowledge representation relative to search in typical AI and cognitive science programs.

In the early 1970s, AI and cognitive science witnessed the appearance of several indigenous and closely related ideas, clothed in varying disciplinary and methodological garb. The state of the art

by the mid-1970s was characterized by a larger granularity in the features of declarative representations in AI. The General Problem Solver had indeed tried to find universal forms of search which were so primitive as to be applicable in all contexts. But the form of its representation was invariably in 'toy' domains, rather than in serious ones. This meant that while search could be relatively efficient, the problems for which it was being clocked were never very imposing. The shift of emphasis to domains rather than procedures and the work on semantic information processing and more emphatically the DENDRAL work in representing domains meant that well-articulated but weak (domain-impooverished) search methods could be felicitously combined with strong domain representation.

The chunks of knowledge entailed in the concepts being put forth were themselves far larger than a decade before. Based on the development of several strategies for blind or syntactic search, the proposal of molar domain schemata in the form of frames, scripts and knowledge bases, provided a research agenda for the rest of that decade. It is of course, only fair to acknowledge that the most high-minded and prescient among the AI researchers had designed list processing programs which sorted information according to attributes, in explicit recognition that the storage unit was larger than the current state of the art could handle. For instance, discrimination nets constituted miniature histories of search for a specific context. Simon considers attribute lists, found in IPL in 1956, to be the Ur-form of frames. That is, in a much more primeval form, these things had appeared in earlier AI artifacts. However, they had lacked the means to instantiate such insights in programs which recognized and properly stored information.

The semantically similar concepts- scripts, frames, schemas (or schemata), are characterized by a high level of granularity, and by recognition that human knowledge representation and storage is habitually formed in high-volume units. This is the case, it is postulated, regardless of the representational primitives. This agenda proved to be appropriate for the first introduction of symbolic representation as technology, as will be discussed in the next chapter. This is a silly paternity battle, since there were so

many closely associated concepts that they seemed to develop coincidentally. As Patrick Winston told an interviewer:

“...the time when people gave up searching for a Holy Grail, a universal mechanism, and began to feel that maybe this enterprise of intelligence would require lots a mechanisms, not just one universal mechanism. Now I think 1970 sort of marks the beginning of new thinking about that. If you look at the GPS and that sort of stuff, I think that was late 1960s. And it was at that time that Newell and especially Simon were making predictions about how smart the machines would be in ten years, not because they were foolish, as John McCarthy once pointed out. This is my recollection of what John McCarthy said once. McCarthy was saying that was not a foolish remark that Simon made; it was an attempt by a responsible scientist to be responsible to society about something he thought really could happen. And what he thought really could happen was that the general problem solver might be deployed on the problem of making itself smarter to produce a sort of chain reaction that would lead to an intelligence of phenomenal capability, which would in fact, had it happened, have had tremendous impact on society by now. So there was poor Simon trying to be responsible in a time when everybody was saying that scientists should be responsible, only to get elevated by his own petard, because something that very well could have happened didn't actually turn out to happen. To this day, Newell I think believes in universal engines. I think he believes that SOAR is a universal theory of intelligence. But I think in the early 1970s, around 1970, some people began to think there was more to it than just one mechanism. Now we have that diversion complete in the form of Marvin's Society of Mind theory, which is a complete antithesis to the SOAR attitude that there is one universal mechanism. So when I said mid-1970s I think I was talking more in terms of collaborations with real scientists. When I talk about early-1970s I think we're talking about a shift in how people in AI were thinking about how AI would evolve” (597).

Scripts

These modes of knowledge representation took granularities of different forms, despite thematic similarities. The concept of scripts, invented by Roger Schank and Chris Riesbeck at Stanford roughly

between 1969 and 1973, and then developed further at Yale, posits the fundamental congruities of much of the human experience (598). Should we consider everyday life rather than the heights of intellectual activity, it seems true that a huge portion of the human experience does indeed consist of following generally similar and indeed 'stock' and repetitive activities: meetings, answering email, picking up the phone and dropping off the dry cleaning, and going to restaurants (the example usually used to illustrate scripts).

These types of episodes are called scripts. Scripts is the central unifying object in an essentialist theory which is referred to as conceptual dependency. Essentialist theories, as this one is, posit fundamental units- atoms, chemicals, countries- of which a given sort of stuff is composed. In pulling apart language, Noam Chomsky, the towering Everest of linguistics during the second half of the 20th century, stated with emphatic rhetorical certainty and little substantive evidence, that human beings operated with an 'innate grammar', fundamental identical features in languages from those of New Guinea to German. An essentialist theory of human events and language boils events down to "ACTs", fundamental building blocks such as moving things, taking them away, and various forms of interacting with people. The scripts or 'CD' (conceptual dependency) theory of language understanding boils the semantics of language down to the expression of these basic acts.

What about syntax ? Schank is credited with the dismissal of syntax with the phrase, "syntax is bunk", which could doubtless make him unpopular in English departments. Schank and Riesbeck's theory appears to be a computational and semantic version of Chomsky's. At an adequately general level, this would seem to work, although given the localness [sic] of human languages historically the absence of some substantive differences seems suspicious. In a rather straightforward parallel to Chomsky, the theory of conceptual dependency posits an "innate grammar" of human events:

" Throughout our research it was our goal to solve the paraphrase problem. We wanted our theory to explain how sentences which were constructed differently lexically could be identical in meaning. To do this we used the consequence of (1) and (2) to derive a theory of primitive ACTs. Simply stated, the theory of primitive ACTs

[a word not acronym] states that within a well-defined meaning representation it is possible to use as few as eleven ACTs as building blocks which can combine with a larger number of states to represent that verbs and abstract nouns in a language. ...We claim that no information is lost using these ACTs to represent actions.

The advantage of such a system is this

1 paraphrase relations are made clearer

2 similarity relations are made clearer

3 inferences that are true of various classes of verbs can be treated as coming from the individual ACTs. All verbs map into a combination of ACTs and states. The inferences come from the ACTs and states rather than from words.

4 organization in memory is simplified because much information need not be duplicated. The primitive ACTs provide focal points under which information is organized." (599).

One might look at these larger chunks of semantic reality either longitudinally, as strings [sic] of procedures; or in static snapshot, as tableaus. Scripts, a sort of knowledge stored in stories, takes the former rendition of reality, Marvin Minsky's concept of frames the latter. Defined in 1974 by Roger Schank, scripts are a fairly self-evident form of knowledge representation. A script is a default representation of a narrative. The representation provides extensive caveats in a form resembling production rules (IF X1, then Y1, if X2, then Y2...), since the values of the state space [sic] at every step may diverge from the stipulated sequence. The narratives of scripts tend to enact nearly-universal human social interactions, such as the ever-repeated restaurant script. The sequence of generic functions of events within the scripts is always reiterated, although the values or forms each of those functions might take differ greatly.

Like production systems, scripts is apparently designed carefully at both the molecular or bottom-up level, as well as from the top-down, or the molar side. Moreover, behind the knowledge artifact of scripts is a theory of cognition and representation. Schank holds out claims of cognitive emulation as well as AI engineering for scripts, purporting that these are the modal unit of knowledge representation. At the minute level of discrete micro-representations, Schank asserted that there are only a limited

number (perhaps two dozen) essential verbs in terms of which all functions within a script may be defined. These individual molecules of which a script was composed are referred to as semantic primitives. Underlying the concept of a limited and defined repertoire of semantic primitives is the idea of universal semantics. Schank's semantic primitives implemented Chomsky's concept of an innate and universal grammar, except that in place of grammar, Schank was concerned with semantic basics.

The scripts structure is designed around material objects, and has a strong semantic sense to it.

It is about what people know about everyday encounters, and how to characterize such encounters. Schank has quipped that 'syntax is bunk'. This AI construct was sort of a computational instantiation of the innate grammar proposed by Chomsky, with a twist. Whether this is a generally universal semantics is almost a moot point because it is so difficult to falsify: this would appear to be as true in the linguistics case as it is in Schank's cognitive science and AI case. (This contribution has apparently remained largely unacknowledged by linguists, despite the useful instantiation of their ideas; Buchanan article). Schank's background as a linguist makes the inheritance of this idea from the world's most famous linguist seem quite probable. Schank's first degree was in linguistics from the University of Texas at Austin; he also studied AI at Stanford with John McCarthy before being hired at Yale (1974). The usefulness of a longitudinal sequential script as a knowledge representation format was first substantiated by the dissertations of Schank and Abelson's first round of Yale students (600).

Frames and Frame Representation Languages

Marvin Minsky's concept of knowledge representation through 'frames', representational structures with generic functions that may be filled with various predicate and object values, bears similarities to the idea of scripts. This concept was first presented in "A Framework for Representing Knowledge" (1974), (601). Minsky asserted that the basic unit of knowledge was not atomic but was based on large, unified declarative objects. Unlike scripts, this general data structure was not necessarily a narrative. The various relevant categories by which events may be described in this

structure are called slots, and the data in slots called attributes (602). According to Minsky,

“ When one encounters a new situation (or makes a substantial change in one’s view of a problem), one selects from memory a structure called a frame. This is a remembered framework to be adapted to fit reality by changing details as necessary.”

A frame is a data structure for representing a stereotyped situation like being in a certain kind of living room or going to a child’s birthday party. Attached to each frame are several kinds of information. Some of this information is about how to use the frame. Some is about what one can expect to happen next. Some is about what to do if these expectations are not confirmed.

“ We can think of a frame as a network of nodes and relations. The top level of a frame are fixed, and represent things that are always true about the supposed situation. The lower levels have many terminals-- “slots” that must be filled by specific instances or data. Each terminal can specify conditions its assignments must meet. (The assignments themselves are usually smaller “sub-frames”). Simple conditions are specified by markers that might require a terminal assignment to be a person, an object of sufficient value, or a pointer to a sub-frame of a certain type. More complex conditions can specify relations among the things assigned to several terminals.”

Collections of related frames are linked together into frame systems. The effects of important actions are mirrored by transformations between one frame and another represent the effects of moving from place to place. For non visual kinds of frames, the difference between the frames of a system can represent actions, cause-effect relations, or changes in conceptual viewpoint. Different frames of a system share the same terminals: this is the critical point that makes it possible to coordinate information gathered for different viewpoints.”

“ Much of the phenomenological power of the theory hinges on the inclusion of expectations and other kinds of presumption. A frame’s terminals are normally already filled with default assignments.

Thus, a frame may contain a great many details whose supposition is not specifically warranted by the situation. These have many uses in representing general information most likely cases, techniques for by passing logic and ways to make useful generalizations.” (P. Winston, Proposal to ARPA”, MIT AI Lab. AI Memo 366, May 1976, p25) (603).

The concept of a need for reference to canonical ‘frames’, in context of which to refer to specific items, arose from the semantic demands of the vision project which the MIT AI Lab undertook during the latter part of the 1960s. Minsky asserts that the perception of scenes, like the traversal of sequences of actions, is undertaken with reference to memories of similar ones. Moreover, this perception is not bottom-up and incremental, but rather draws upon larger compilations of attributes and predicates typically encountered under the larger auspices of any given frame.

Frames, Schemas, Scripts, KRL, and Semantic Representation Units

An idea whose time has come may appear in a number of forms at roughly the same time: for example; the nation-state; different versions of fascism, and personal computers with ersatz operating systems, and even the electronic digital computer, all appeared at roughly the same time. This is what happened in this instance. A handful of superbly smart people pursued generally similar ideas. The common denominator to all of these sorts of AI KR structures was their granularity, that is their assertion that data structures were large rather than small, and their accounting for narrative and causal structures. Between Minsky and Schank, one sees the sources of a good portion of the Ph.D. output in AI starting in the 1960s (and 1970s for Schank). Avron Barr, general editor of The Handbook of AI, asserted that Schank himself oversaw half the dissertations in the field of AI (604). Moreover, Minsky’s influence has clearly always been enormous. Thus, their ideas had tremendous force, in a generally similar way despite the different areas of applications of scripts and frames.

Like Schank, Minsky asserted the primacy of large units of knowledge, although frames were not explicitly chronological and did not pay homage to Chomsky. In the case of frames, the

underlying reference was not to Chomsky but to scene analysis in the vision research undertaken in the MicroWorlds project.

Thus, it is not surprising that the work of Minsky and Schank was so richly evidenced in the works of other people during the late 1970s. Minsky's original article had been a 'thought piece', and does not propose a computational formalism. 'Frames' became the prevalent appellation for KR of this variety, and various frame-based programs were created during the next two decades. Minsky inspired applications based on frames in the several years after the initial appearance of his paper.

While the concept of Frames was tremendously generative, it is not clear whether the name of frames was useful. It was messy in at least two ways. First, it sounded a great deal like the Frame Problem, the logistically and philosophically messy issue of how a problem solving system would know where to stop in examining its knowledge base. This problem was, as we saw earlier, apparently first identified by John McCarthy in his "Advice Taker" paper in the late 1950s. Second, Minsky's frames irritated Herbert Simon. Simon also found 'his nose put out of joint' by Minsky's claim that frames actually set off the object orientation development in software engineering (605). Simon objected to Minsky's usage of the term 'frames' at all, since it clearly confuses the frame representation issue with the Frame Problem. Simon asserted that he and Newell actually invented attribute lists, the Ur-concept for frames, but that the technical hardware limitations of the 1950s imposed severe limits on what could be done. This simply seems to be another case of both Minsky's catching the zeitgeist at the ideal moment, and of Simon's inventing various ideas before they could be implemented technically.

At about the same time as the canonical Frames paper was written, the idea of schemas, a form of semantic net, was introduced by Bobrow and Collins'. Published in their edited volume, Representation and Understanding (1975), does not seem to differ much semantically from Minsky's.

To further confuse matters, psychologist David Rumelhart, then at U.C. San Diego, published Parallel Distributed Processing

(commonly known as PDP) in 1973. Rumelhart's original work on this theme appeared earlier than the famous PDP and was published in Bobrow and Collins' aforementioned Representation and Understanding (1975). He proposed a concept also labelled schemas or schematas as the unit of human information processing. 'Schematas' in this connectionist architecture reflect 'emergent states', rather than persistent objects. The label is used to designate KR more volatile and transitory than those in typical knowledge representation.

Fielded Production Systems

Viable Production Systems at Stanford and CMU

The production system formalism was composed at a primitive level of 'condition-action' or 'if-then' rules. This was an efficient way to navigate even large and detailed knowledge bases because of it specified means by which to proceed at a minute level. Newell and Simon's original intention for this knowledge representation had been for cognitive emulation, but the system was used to great advantage for problem-solving in technical areas. When used to provide a means of making inferences in established state spaces, the emphasis of production systems turned from the 'weak' or general features of representation of information- the cognitive science orientation for which production systems were originally designed- to the strong or domain-oriented pragmatism. The production systems formalism, turned to this technical usage as a form of artificial or synthesized inference-making, is called an expert system (Edward Feigenbaum's term), or an inference engine (Randall Davis' term).

Meta-DENDRAL

We discussed the origins of Dendral in Chapter Six. The Dendral Project, renamed the Heuristic Programming Project (HPP) in 1972, maintained its generativity. Dendral's contribution had been its ability to solve complex scientific problems and embody the knowledge therein clearly. Its distinction is based on the fact that it managed to actually solve the problem in the field sought, in a way that plausibly resembled the human experts' methods. In

introductory AI courses, one may hear a criticism of Dendral and Heuristic Dendral, alleging that it is not an “expert system anyway. This is because its first step consists in generate and test, and this is followed by a filtering through using constraints in the form of production rules. It is interesting that such brainstorming- perhaps the most appropriate colloquial term for this form of problem-solving- resembles the human creative process. At the same time is routinely referred to as inadequate and risking combinatorial explosion. Heuristic Dendral, described in the previous chapter, could be labelled a production system due to its usage of condition-action pairs to narrow down the profusion of possibilities of the initial generate-and-test stage. It precluded computationally intolerable combinatorial explosion by only testing syntheses which met established constraints.

The final sibling, Meta-Dendral is a very different program. Meta-Dendral is the mother of all expert systems. As progenitor, it is like others; its basic outlines are implicitly honored by everyone, but its particular techniques are only faintly in evidence in any particular system (606). It conducts inferences based on Heuristic Dendral’s database. This feature explains the title; the program is a “meta-program”, since no new semantic knowledge is actually introduced. The prefix indicates that the named program itself is being addressed further. It is not an adjective describing the means of problem-solving employed by the program (as “Heuristic Dendral is”). Inferences conducted upon the substrate of Heuristic Dendral’s knowledge base constituted automated, or at least highly disciplined, theory formation. Meta-Dendral’s rule formation is conducted based on Heuristic Dendral’s raw data, specifically six to ten pairs of input-output sets (i.e., training sets), illustrating the patterns according to which new compounds will actually fragment. These subsequently result in rules, formatted according to a predetermined syntax. The program is capable of integrating ‘late-breaking’ and incongruent knowledge by the inputting of more training pairs (607).

This particular usage of the input-output information is germane for at least two reasons. First, its results are symbolic and consist of condition-action pairs rather than the garnering of statistical probabilistic data. It maintains its identity as a knowledge-based

and semantic program rather than a statistical one. Second, the retroactive imposition of a syntactically regulated rule base segments the data itself from what is done with it. Earlier versions of Dendral had been criticized for their (putatively) loose structure, since they had consisted of unsorted and somewhat lengthy bodies of LISP statements. The final Dendral program was more parsimonious. This makes Meta-Dendral a 'Missing Link' in between earlier programs in which the semantics and syntax could not be separated, such as MicroWorlds' SHRLDLU, and the expert or knowledge base system, in which methodological skeletons (that is, production systems as a KR problem-solving format) could be separated from various similar knowledge bases in isomorphic domains.

The MYCIN Experiments

Allen Newell called MYCIN "the original expert system". MYCIN is an applied production system concerning a deep and complex domain, specifically the diagnosis of illness in internal medicine. More precisely, the program reviews the symptoms of infectious blood diseases, and then suggests diagnosis and treatment (608). It provided plausible interpretations of various physical symptoms in this domain. Where Dendral was the product of many years of research with a large supporting cast of professors, MYCIN is described by one of its authors, Bruce Buchanan, as an effort to come up with something from the bottom up, with almost no money at the start- Buchanan calls it "bootlegged". Later on, MYCIN and related medical diagnostic work received NIH and NSF grants (609). Stanford MD-Ph.D. Edward Shortliffe was the other of the pair who wrote the program- Mycin was his doctoral dissertation (1976) (610). The system performed quite successfully in clinical evaluations: "When a panel of experts evaluated the performance of several different agents, including medical experts, interns, and MYCIN, MYCIN's performance was judged as good or superior..." (611).

Structurally, at both the primitive or atomic (molecular) and a global (molar) level, Mycin differs from Meta-DENDRAL. It relies upon the classics, so to speak, and upon a highly parsimonious programming protocol. All statements concerning semantics are expressed in the

syntactic form of production rules. Within the latter statements, relations amongst variables are in Boolean form (612). Unlike any of the Dendral programs, MYCIN expresses its diagnoses in probabilistic terms. MYCIN assigns Bayesian probabilities to different potential diagnoses, rather than statements of certainty (this is referred to as the confidence factor, or certainty factor (613). In the context of medical diagnostic system, this seems more congruent to the tentative way that physicians actually state their conclusions.

The Meta-Dendral program (if not the other Dendral programs) had been justifiably labeled a production system based on its creation through inference of 'second-order' rules concerning their data. However, MYCIN was based entirely on production system rules, and more justifiably so. Critics of the Dendral experiments, such as they were, had also complained about the relative inaccessibility of the knowledge base in the latter programs. The latter was congealed into a dense thicket of LISP statements. Mycin was a much shorter program, its data being more concise due to syntactic discipline. Moreover, it managed to embrace a good volume of information: one thousand rules and twenty class names (614).

The linear flow of problem-solving in Mycin is intended as an act of reverse engineering, reiterating the analytical processes of experts, specifically internists, in diagnosing illness. The internist consulting MYCIN enters data concerning specific symptoms of a given patient, into the program. Because the system uses production rules, appropriate terminal points (or diagnoses) are considered according to the right-hand implications included in the program. The pursuit of childless nodes is precluded by the pruning built in with these right-hand clauses. This strategy is called backward chaining. In colloquial language, we might think of it as a process of gradual elimination of intermediate options when faced with a known goal. It could also be seen as a somewhat more refined form of means-ends analysis.

This knowledge architecture would certainly appear to be more economical than its predecessor. It does not even risk combinatorial explosion, and all nodes which it chooses must be correct. On the other hand, MYCIN's task is diagnostic rather than exploratory,

which may render the commonly made comparison irrelevant. It did not address problem-solving, though, nor did it generate hypotheses about plausible combinations of primitive elements. Instead, it employed pattern matching as a means of diagnosing selected medical circumstances. It did this in a structurally disciplined way. This consisted of combining a fundamentally simple low-level architecture (based, like Meta-Dendral on productions) with a very rich understanding of the higher level relations within that data. A separation of the rule base from the knowledge base was going to be necessary for any abstraction of one tool for a specific domain to any other domain, even one that was closely similar. That is, a tool that was more widely applicable would require such a more abstract heuristic structure versus domain substrate separation.

Subsequent Related Work

Despite its landmark status, the MYCIN program was one of a continuous line of developments in knowledge representation at Stanford during the late 1970s and early 1980s. These took place iteratively and rapidly, some even within the same year, as if each new program was an answer to the possibilities implied in the previous one. A small but productive cohort of bright computer science dissertation writers- who this author labels 'the Teknowledge Generation' after their leadership in the introduction of commercial expert systems several years later- along with several members of the computer science faculty (especially Buchanan, Feigenbaum, and Lederberg), was responsible for this rapid progress.

Further innovations were introduced by TEIRESIAS (1976), EMYCIN (1979), PROSPECTOR and KAS (1979), PUFF (1979) and RLL (1981). These turned the initial invention of parsimonious mechanized inferencing into a crafts skill for AI-trained computer scientists. By rendering the structure of the rule base, or 'inference engine' uniform and extensible, and making this applicable to a variety of similar knowledge bases in different domains, this form of problem-solving could be applied quite widely. All of these fit under the genre of 'inference engine', the semantics of which terminology are worth considering in themselves. Randall Davis coined the term in

the course of preparation for a DARPA review panel (615). The name is both suitable and attractive. The mechanical versus cognitive or biological designation makes the program seem workmanlike and practicable rather than endowing it with mystical animistic properties. Moreover, the semantics of the term are good: an engine implies useful production, active processing of some sort of input, and relative clarity and technical maturity of such processes. Inferencing is of course what such production systems were intended to do. The term leans heavily toward technology, and even hints at economic usages. The transformation of the academic culture into a much more technically and less intellectually-oriented one appears to be evident in this wording. Expert system and knowledge base system, which have become more widely used for reference to an applied production system (and various variations thereon), both use the vague term 'system' and specify expertise (somewhat overreaching themselves) and the relatively static reference to knowledge. While it has not been pursued, inference engine remains the superior nomenclature.

The first two of these programs appeared in the course of ongoing collaboration between Shortliffe and the other early knowledge engineers. Randall Davis, director of MIT's AI laboratory during the 1990s, produced TEIRESIAS as his 1976 doctoral dissertation. The program's title is the name of the blind sage in Oedipus Rex. Buchanan, thesis advisor to Davis as well as Shortliffe, had a background in philosophy and Classics which he has delighted in drawing upon. The intended usage was as the first knowledge elicitation program, or auxiliary 'interrogator' to help physicians interact with MYCIN. Created to facilitate the transfer of knowledge from the domain expert to the program, TEIRESIAS did not request the exoskeleton of the knowledge the form which was required by production systems.

Almost concurrently, William Van Melle at the HPP created EMYCIN, a domain-independent version of MYCIN (616). EMYCIN's adaptation consists in removing the object attributes from the rule base of MYCIN. This made MYCIN a skeletal system that could be adapted to various other circumstances, to especially diagnostic and troubleshooting applications. Such applications extended to commercial 'shells' such as TI's 'Personal Consultant' software

system. Other applications were closely comparable to MYCIN and were in the medical domain. PUFF, which analyzed data on patients thought to have lung disease, was an HPP project to produce an analogue to MYCIN for a different domain (617). PUFF could diagnose lung disease. This was also the first program constructed using EMYCIN. This program was remarkably successful outside the laboratory:

“ Once the rule set for this domain had been developed and debugged, PUFF was transferred to a minicomputer at Pacific Medical Center in San Francisco, where it is used routinely to aid with the interpretation of pulmonary function tests. A version of PUFF has been licensed for commercial use.”(618)

The sorts of domains for which production systems inference engines were appropriate were usually technical and well-ensconced in codified knowledge. Not only should the information be unambiguous and complete at the level of molecular discrete facts, but the overall diagnosis had to be very clear to begin with. One might say that such domains were ‘crew-cut’ rather than hairy domains. Such domains were quite well-behaved with regard to hierarchical classes of relations, with the inheritance structures consistent. The caveats to diagnoses in search in such knowledge bases were intrinsic to the knowledge itself rather than to the representation of that knowledge. In the medical cases, even extremely well-trammeled domains could yield only probabilistic, rather than certain, answers to diagnostic questions. This problem was alleviated by the confidence factor and the Bayesian probabilistic responses. Domains which were ideally suited to such production system representation include mineral prospecting (Prospector), medical subspecialty diagnostics (PUFF, OncoCin, MYCIN and NeoMYCIN), structural engineering (SACON, Structural Analysis CONSULTANT (619), and customized computer sales configuration (XCON). IPTO subsidized applied production systems, first in volumes ample to cover many graduate careers during this period.

Two other programs extended the search capabilities of applied production systems further. Prospector and KAS were related in a dyad parallel to the one remarked upon in MYCIN and TEIRESIAS

several years before (620). Various Stanford HPP researchers, working at SRI in cooperation with Elf Aquitaine, invented the Prospector program (621). The program's purpose was oil prospecting, or rather, the analysis of geological features of given locations, and prognosis as to the likelihood of finding oil. Using the constraints imposed by a dozen often orthogonal knowledge bases, it struck oil at better-than-chance intervals. Whereas MYCIN had used backward chaining to limit search by pruning away unproductive nodes, Prospector used both backward and forward chaining. In order to 'speculate' as to plausible diagnoses, the program could implement forward chaining. Such speculations could be pruned by the constraints established by consequent clauses further forward in the KB. Like the medical models, it used certainty factors -0.5 to $+0.5$ to indicate its prognosis as to the likelihood of given conclusions. Even where an 'IF' could plausibly lead to a 'THEN', the likelihood of this connection being followed was given a coefficient.

The Knowledge Acquisition System, a more advanced version of TEIRESIAS, was developed along with Prospector (same original authors and references). Like Teiresias, KAS does not assume that the expert 'knows' in axiomatic form the often tacitly held structure of knowledge available to the domain expert. In this sense it is as much an epistemological tutorial as it is an interviewing program strictly speaking. It iteratively solicits and then tests IF-THEN antecedent and consequent clauses from the domain expert, until all facets of a new structure are completed.

XCON and Caduceus

The respective development of heuristic inferencing programs, applied production systems ('expert systems'), and expert system builders at Stanford during this period took on the aspect of an intellectual movement (albeit, a small movement). This does not appear to have been the case at Carnegie-Mellon, where Newell and Simon were simply more oriented toward science than toward technology. Since expert systems are a form of production systems, such a segue into this applied form followed by commercial exploitation might have been expected, but it did not materialize. MIT's intellectual tradition was highly disparate, as has been

established earlier: and generated neither applied production systems nor commercial expert systems. We must issue this caveat, since the frame formalism invented at MIT was introduced into expert systems a number of years later.

Thus, it remains somewhat perplexing that the very first applied expert system was created by John McDermott of Carnegie-Mellon. McDermott was at the time an assistant professor. Built in CMU's preferred OPS5 language, the eXpert CONfigurer was built on the fly when McDermott was assigned the task of computerization of the configuration of VAX mainframes for the Digital Equipment Corporation (622). Still in use decades later, XCON is the most durable and pragmatically successful expert system, and it is significantly larger than MYCIN ever became, with six thousand rules and one hundred class names (623).

MOLGEN, Units, KEE, and the Merger of Frames and RBS

Since production systems and frame formalisms embodied complementary advantages, it is not surprising that there was a merger of these two forms of representation. For relatively homogenous classes of tasks, frame representation standardizes semantic categories of objects and procedures which must be attended to. For problems which are less intrinsically hierarchical in design, and therefore somewhat more realistic, production systems insure the solution of all features of complex problems. A production system is a search engine, which a frame is not. And a frame system is a basis for knowledge representation, potentially effusive, which a production system is not.

The project which put into effect this merger was the Molgen project, a cooperative effort centered at the KSL with loose borders connecting it to Xerox PARC. This became the centerpiece of the technical basis for commercial KBS soon afterward. In Molgen and its related systems, Peter Friedland and colleagues developed an applied production system which was self-consciously designed to mimic human problem-solving protocols. An applied production system solved problems either from the bottom up or according to a specified control structure. Higher-level control structures recursively circumvented the search further below if inputs

contradicted constraints. This constituted efficient problem solving per se, but not cognitive verisimilitude. The Molgen cohort introduced increased resemblance to structured, syntactically preordained scientific experiment design, and hence the referral to it as a frame-production system hybrid. Of the two preliminary 'pre-Molgens', the first fit the design of the pseudo-expert system more closely to the scientist's putative protocol:

" One of the systems introduced the concept of "skeletal plans", which are abstracted outlines of experiment designs that can be applied to specific experimental goals and environments." (624).

The second introduced planning constraints to preclude failure nodes for the given scientific field. Molgen and sons led directly to the techniques which were decisive and influential as commercial AI. Molgen at KSL branched out into Units at PARC. Units, created by Mark Stefik- of whom more later- was the AI project which led directly to KEE.

Convergent Ideas outside of Artificial Intelligence

Structured programming:" a set of programming techniques for implementing program and control abstractions in a hierarchical manner. Its purpose is to make programs more readable, less error-prone and easier to maintain. three techniques in this category are modularity, top down design and the use of structures control constructs". (625).

The chief notable feature in the developments in AI languages and artifacts at the turn of the seventies was the shift to semantics and to much larger representational units. It is intriguing to observe parallel but quite independent developments in other fields. First, the 'Structured Revolution' in computer languages greatly enhanced the discipline of the discipline. Edsger Dijkstra's seminal paper, "Goto statement considered harmful" (626), berated programmers for their improvised ad hoc programming style, epitomized in their usage of 'goto' when they ought to have actually defined an object.

Dijkstra, a famously ascetic Dutch theoretician, introduced the term 'structured programming'. The computer languages Algol 68 and Simula67 (developed variously by committees and by European researchers), concurred with Dijkstra's ideas. Both languages proposed a virtual frame for standardized canonical structures for computing language terminology and procedures. Simula's classes and Algol's blocks both defined persistent objects and thus represented the beginning of semantics and syntactic operations localized to particular classes of data. This constitutes a close correlate to the domain centrality which was then rearranging the entirety of Artificial Intelligence programming. Dijkstra and the aforementioned designers of Simula and Algol were concerned with computer language design and pure theory, not with cognitive science, but in certain respects the isomorphism is remarkable. It is this movement which led several years later to the early languages of object-oriented programming (Pascal and Smalltalk were directly inspired by Algol and Simula respectively) (627).

Dijkstra himself, a theorist at a Dutch research institute ("THE"), was not affiliated with the AI cohort (628). In this sense, structured programming and OOP were genealogically separate from AI despite the general field of computing. However, the common state of instrumentality in computing at the time makes these developments concurrent and in a sense capable of taking place at all because of common features.

It is interesting to note that the shift to domain-specificity appears to have appeared independently in fields other than AI as well. As these developments were taking place in AI, grounded theory was simultaneously becoming the *modus operandi* in the social sciences, and a distinct and different epistemology for knowledge *per se* in other fields.

A further structurally similar but more clearly independent intellectual secular trend was the birth of several "domain-based" movements in the social sciences. In Marxian social science, an increasingly vogue approach to social phenomena, a British and American social-history-approach eclipsed the mostly French 'structuralist' analysis. The same was true of the field of history itself with the appearance of the *Annales* school of social history

and increasing usage of measurement in history. The functionalist take on sociology, introduced by Talcott Parsons in close congruence with Cybernetics, had been heavily top-down and abstract, emphasizing the homeostatic functions of entities in society. This was replaced by emphasis on conflict and by 'grounded theory' which required empirical research. These were, and mostly remain, highly orthogonal and separate intellectual communities, the denizens of which do not appear to interact a great deal. It is curious that the 'local' or bottom-up answers reached independently by these distinct groups were so similar in domain-specificity.

Blackboard Architectures and the Cautionary Tale of ARPA-SUR

A blackboard architecture is a knowledge representation formalism which holds heterogeneous types of data. The latter are attached to procedure calls, rendering control fairly distributed. This sort of formalism can process information from highly varied sources, and can address complex objects or engineering situations. This has been useful for engineering and military uses, as well as applications which generally address problems in large artifactual structures. The latter is true of a frame as well, although BBA differs in that the work was not a putative commentary on psychological architectures. Instead, blackboard architectures has been purely 'knowledge engineering'- that is, an effort to solve engineering problems without any particular reference to psychological phenomena. It differs from a script in that the latter is explicitly narrative, and takes on the much larger epistemological claims of the theory of conceptual dependency.

Blackboard architectures (BBA) are interesting in part not because they are new, but because they aren't. Instead, they are actually old- a classic idea. The concept of a blackboard architecture originated in Pandemonium, the relatively flat agency architecture designed by Oliver Selfridge in 1960. Pandemonium had differed from the other problem spaces being explored at the time. These comprised, generally, hierarchical searches through a vertically rooted graph. In Pandemonium, in contrast, the knowledge base was diverse- "a blooming, buzzing confusion", to use William James' felicitous phrase, and thus functionally independent units could

solve problems incrementally. This is a concept that is computationally and intellectually intrinsically appealing, and because of its suitability for engineering and distributed problems, it was ancestral to other forms of AI. Pandemonium was the Ur-BBS, so to speak, but gave rise to other things as well. Selfridge's program, which influenced Newell through his friendship with Selfridge, was influential for production systems, too (629).

Selfridge had inaugurated this form of knowledge architecture with a concept that was intermediate between connectionism and the general problem solver type of research that comprised a good deal of the most sophisticated AI at the time. But Herbert Simon seems to have restored interest in this KA by drawing attention to it. In the mid-1960s, Simon explicitly articulated the blackboard model, again in a theoretical essay (630). Discussing human problem-solving protocols, he refers to the habitual generation of sub-goals, the incremental interim solutions to which are held in a 'blackboard' in 'permanent or relatively long-term memory', during the course of addressing the problem's subgoals. This usage resembles the one currently presented as a blackboard architecture. This, in turn emerged from a combination of the interest in the flat topology just presented, and a novel problem presented to the AI community.

Late in the 1960s, IPTO determined that it would finance a major project in SUR. The IPTO assembled a distinguished committee, including notably Allen Newell, J.C.R. Licklider, and Raj Reddy, to establish performance specifications for the contractee projects. This committee's recommendations were demanding, if somewhat flexible (631). Performance criteria for the systems included the emulated comprehension of normally spoken, continuous speech (i.e., 'connected speech'), a vocabulary of one thousand words, and a "reasonably fast" response time, "with less than ten per cent error" (632). The specification of a "reasonably fast" response seems somewhat vague, if the other elements do not. Moreover, neither the nature of the "constrained" syntax nor any particular words in the lexicon were specified. The performance specifications did not actually impose the full difficulties of the speech understanding task upon the contractees. Still, these criteria considerably surpassed the state of the art: the emulation of comprehension of connected speech was not possible at that time.

Moreover, the recommendations were not at all naive: the Handbook observes that typically AI projects stipulated design objectives rather than (perhaps more difficult) performance specifications, before they were undertaken.

The opportunities afforded by ARPA-SUR monies financed a number of projects in speech recognition, and led to a knowledge formalism as well. In this sense, ARPA-SUR parallels the MicroWorlds project, which began as a problem for vision but inspired work in novel knowledge architectures as well. Both of the projects started as “applied” projects in the areas of sensors and effectors, and then turned into more general theories of knowledge architectures. The parallel does not follow in the attribute of the source of the project’s initiative. Minsky and Papert arrived at the MicroWorlds puzzle of their own accord, while ARPA-SUR was proposed by ARPA’s Information Projects Technology Office.

The speech problem is quite heterogeneous. The Handbook of AI specifies at least nine distinct and unrelated elements, all of which pertain to the domain. These include phonetics (the representation of physical features of sounds in words), phonemics (rules concerning variation in pronunciation; prosodics (rules governing the usage of fluctuation and stress in sentence intonation); morphemics; (rules concerning the way that units of meaning, or morphemes, are combined to form words); pragmatics (the coherence of conversations, regardless of the legality of the syntax of sentences), phonemics (the abstract representation of sounds in language), and the interpretation of allophones (phonemes as they actually occur in language; more of these, since these are all variations in phoneme according to context) (633). Other issues, such as semantics (the meaning of utterances) and syntax (the legality of sentences), are part of natural language understanding as well. It is noticeable that the problem of natural language understanding as we have seen it so far significantly constrains its domain, and omits several of these further challenges in the process. The semantics of problems to which this is well-suited are typically highly heterogeneous databases with orthogonal elements. (Thus the blackboard data structure does not imply expensive redundancy, and the expense is in increasingly cheap programming instead).

IPTO diversified its risks by assigning the contracts to several different institutions. The list included mostly veteran ARPA players, such as Bolt Beranek Newman, Carnegie-Mellon, SRI in collaboration with the Systems Development Corporation, and Stanford (634). Such major contractors as BBN and CMU produced more than one system each. CMU in particular is notable, as it produced HEARSAY-I and DRAGON systems between 1971 and 1973, and then HARPY and HEARSAY-II programs by 1976 (635). Hearsay was the first BBS, and its immediate successor, a later-discontinued project called Hearsay II, was critical to the development of BBS. The former ARPA-SUR project recalled previously articulated themes in AI, and especially the flat and heterarchical domain addressed in Pandemonium (see earlier in this chapter). It is not surprising that these themes had been addressed by Reddy's advisors. Simon apparently drew Reddy and Erman's attention to blackboards during HEARSAY's design (636).

HASP, which was developed by Edward Feigenbaum and H. Penny Nii in the mid-1970s with significant influence from HEARSAY-II, fits into the designation of a blackboard system in a non-SUR domain. Feigenbaum and Nii's military contract concerned the domain of ocean surveillance, which apparently bore certain isomorphisms to the speech domain. Due to the military domain being investigated, Nii and Feigenbaum were not yet permitted to actually present the substantive topic of their contract work. Thus forbidden to be more specific, they used an analogy to describe the different knowledge architecture features of a comparable object, (a useful exercise in induction as a means to meta-knowledge, as well as amusing). The object of the analogy is idiosyncratic and original. This is certainly the only instance in which koalas, the reclusive Australian marsupial, are ever mentioned in any literature relating to computer science !

Subsequently, the concept of concurrent computation using a diverse knowledge base was taken up by their colleagues, detached yet again from the speech recognition and understanding origins. We should also observe- as we saw in Chapter Eight- that the research funding cut-backs of the mid-1970s rather arbitrarily cancelled the ARPA-SUR monies, to the chagrin of many involved.

Conclusion

Both that AI had happened and how it had happened were underdetermined, especially when looked at from the perspective of the mid-1950s. The understanding of intelligence circa 1956 had been practically entirely derived from Cybernetics. It had consisted of little more than the borrowed concept that intelligence was goal-directed and entailed iterative adaptation of an entity to the feedback from its environment. However, we should acknowledge subsequent influences from the older tradition of Associationism in cognitive psychology and other psychological schools. At the same time, cognitive psychology was rejuvenated almost at the same time as AI was born, and this makes the progress that AI manifests especially impressive during its first few years. A few dozen people (albeit helped by an almost equally small cohort in cognitive psychology) advanced Cybernetics' precepts tremendously from the mid-1950s' point of departure.

During the latter part of the two decades chronicled in this chapter, moreover, AI had asked questions that it could not answer, and managed to organize the research questions and empirical methods to explore, if not the answers, issues of the nature of intelligence. The field had developed a permanent inventory of difficult questions with which to interrogate itself. Even an admittedly incremental approach in that it was low-road and tried only to consider distinct facets of intelligence was an uphill battle. Finally, prospects for AI's future were enhanced by changes in the instruments for experimentation. Complementary hardware technology had been established (not entirely for AI, but certainly to its benefit) which made the research that could answer AI's questions more than theoretically possible.

At this stage of development of AI, one might even see the demarcation of negative heuristics as a useful if severe epistemological device. The establishment of the idea of levels of explanation in different forms of information processing (from the neurological/ biological through the cogitational) had established separate and distinct examinations of intelligence itself. The intelligence from the bottom-up investigations were necessarily

conducted under separate experimental auspices. Unfortunately, particularly before 1985, these were often seen as hostile or opposed to Classical PSS AI, rather than as complementary to it (as was demonstrated during the Perceptrons Connectionism debacle).

But within Classical AI itself, research in different cognitive modalities often resulted in cross- fertilization rather than border conflicts or outright incursions. The vision research which took place as part of the MicroWorlds projects figures in Minsky's concept of frames (and later in the 1970s, the challenging semantics of speech understanding inspired blackboard knowledge representation formalism).

Features of the extraordinarily rapid progress of AI during this period included tremendous accretion of size or granularity in semantic units studied. From the vantage point of consideration of the chronology of a science's life, this progress to much larger-scale models was attained in quite precipitous time. The taxonomical work concerning distinct problem-solving methods, usually blind and syntactic ones, could have been a point on which the field became stuck, and lost both adherents and new entrants. But this potential infinite plain was succeeded by further assaults upon genuine heights. The several knowledge representation formalisms which appeared around 1970 were evidence of this methodological maturity, at a level of problem-solving and search formalisms and languages, as well as a larger coherence of vision sufficient to undertake such higher-scale models.

Chapter 14. Conclusion: the Status Quo in 1980

Introduction

The immense turmoil and technological optimism of the era initiated by the early 1960s gave way, nearly two decades later, to a highly pessimistic epoch. The late 1970s was by far the least militarily bold, most pacifistic, most anti-government, and most anti-business and anti-entrepreneurial (simultaneously) public culture in the second part of the Twentieth century. As we will see in the third and final volume of Building the Second Mind, every

aspect of this political, cultural, and economic climate was soon to be reversed.

Between the Past and its Antecedents: the Gap between 1961 and 1980

Despite the public sentiment, the progress of the two decades bore out the optimistic rather than the pessimistic point of view of AI. This is notwithstanding that there was a great deal of journalistic hay to be made with naysaying, a given because there always has been, and always will be.

This book has covered the historical narrative of the development of AI during the 1960s and 1970s. As we have seen, there was plenty of territory in this itinerary.

This included the commercialization and elaboration of possibilities of the integrated circuit, extraneous to AI but consequential in increasing opportunity to all computing. ARPA, also, was not strictly speaking intended as a vehicle for AI but might as well have been. Other forms of progress in computing were ancillary to knowledge engineering strictly speaking. But complementarities were such that AI could not have had nearly as much progress without it. These included the development of timesharing, input-output improvements such as the keyboard and CRT screen, the conceptual development of the computer utility and operating systems which would facilitate programming for AI among other things. Cultural attacks on AI did not do the field particular damage. In the area of pop culture attacks, were not sufficiently thoughtful to provoke further changes. This would not be true of the later critiques by John Searle (philosophy-based and Pamela and Paul Churchland (neurology-based), but this statement as to the innocuous nature of attacks was still true of the 1960s.

The achievements of the 1960s were seemingly inordinately productive, because of the sheer timing of the field. As we saw in the first book, the earliest successes of AI were hard-won, and were indeed like pulling teeth. Toy problems such as cannibals and missionaries were useful because they could be solved by intuition. This meant that the steps in their solution were relatively simple to elucidate, and then to infer more abstract and universal steps in

heuristic problem solving. The difficulty of the initial illumination of the steps in a particular problem solving heuristic meant however, that it would be much easier to reiterate another problem-solving sequence once those had been understood. These initial successes were hard-fought and problematic at every stage in the late 1950s.

The Visage of the Future

To appreciate the enormity of the change that was going to occur with relative proximity, we should take a swift glance back at the past, which too easily becomes stale the second it shifts dramatically. Much of this history has been the story of the Cold War, and, its ongoing and crucial subsidization of every aspect of computers. This unbearable state of affairs endured: the Cold War was not over until it was over. The Cold War would continue through the 1980s, only really ending with the bloodless 1992 collapse of the Soviet Union. However, the corner was being turned as this book concludes- the USSR was increasingly on its way to giving up, and the USA was increasingly eager for the military components of the struggle to cease.

The November 1980 election of Ronald Reagan, whose domestic policies evoked endless derision as he attempted to weaken the American welfare state in the name of invigorating the American character, offered an aperture to a new United States. This one would attempt to conclude the Cold War through diplomatic detente and through a massive military buildup. Moreover, the new USA would be far more entrepreneurial and business-oriented. Whatever his considerable weaknesses, Reagan succeeded in both of these endeavors.

This glance at the past gives us insight into the future, when the development of computing generally and AI specifically would be responsive to commercial demand, both from a growing general corporate computing market, and from an exponentially exploding and still unanticipated consumer end market. As of the late 1970s, the personal computer was still an odd and exotic hobby item. This was about to change. However, distributed and networked services for business were becoming more prevalent, and this meant a growing market for more usable and attractive end user

applications. Further contributions along the course initiated by DEC computers and studied at the very start of this book would lead to DEC workstations, UNIX machines and LISP machines. Demand pull for applications, cheaper processing and memory, and certainly more aesthetically attractive computers would intensify during the 1980s.

The interaction between AI and commercial computing would be amplified by the staggering and unforeseen boost in the field of computing from popularization of the field, all of which was unforeseen as of 1980. For AI, this would mean vast new commercial markets at first for corporate purpose, in the form of expert systems in the closer future. Later commercial possibilities for AI would include data mining, CRM, voice synthesis and recognition, facial recognition, robotics, and graphics. And, finally, AI hype would truly come into its own, at first with the welcoming of the expert systems overkill of the early 1980s.

But that is another story.

Chapter 15. Acknowledgements

Wandering through the University of California at Berkeley campus during graduate school, the author found herself in the back of an introductory lecture on Artificial Intelligence computing in Cory Hall. As is her manner, she took notes in the form of a cursory list, which turned into far more. A little list can be a dangerous thing. As we know from the history of AI and its predecessors, lists are often at the start of things- commandments, dictionaries, encyclopedias, Cyc, LISP and list processing languages. Learning more did not satiate but made me hungrier: definitions provoked more questions- understanding the terms being used, the development of central problems in AI, the software projects being discussed, and their provenance and creators and chronology. The itch to know more could not be scratched away.

Building The Second Mind originated in the author's doctoral dissertation on the introduction of commercial expert systems in the 1980s- "Developmental Characteristics and Spatial Formation in the Commercialization of Knowledge Base System Shells, 1975-

1991" (May 1993, Department of City and Regional Planning). As is typical for dissertations, if not for this author's intellectual compulsions, that work stayed close to its focus. This was the commercial introduction of expert systems to the turbulent world of software applications of the 1980s, their technological adaptation to that environment, and their mixed success in that endeavor. Stanford, CMU and MIT were considered insofar as their differing intellectual environments produced distinctive commercial cultures.

I did not wish to let go of this issue, and it did not want to let go of me either. In a fine example of emergent functionality, the story took on the proverbial life of its own, and its topics grew into the past and the present. The nucleus of the original thesis concerned the years between 1975 and 1991. In *Building the Second Mind: 1956 and the Origins of Artificial intelligence computing*, that chronological period was augmented by another 'nuclear' narrative concerning the prehistory of AI as an idea.

For a few years, the book was preempted by the consummately important details of caring for children. Unable to say "We're golden," "go big or stay home", or "let's use up all the venture capital money and then close down the company", the author felt like the last scholarly nerd on Earth amidst the financially opportunistic, if brilliantly creative, entrepreneurial environment of the Bay Area at the time. During the Naughts, I developed a nostalgic cargo cult of my own return to my book. In mid-2010, said children were finally more independent, one going so far as to move to New Hampshire to demonstrate this point. The author hopefully took up the project again. Returning to this existential task worth doing was immensely gratifying: There's no place like home.

A list of people who have contributed to this work, through conversations, interviews, or classes I took, includes:

Mie Augier, Avron Barr, Jennie Benton, Daniel Bobrow, Tim Bresnahan, Lee Brown, Bruce Buchanan, Manuel Castells, Marie Desjardins, Paul Edwards, the late Robert Engelmores, Adam Farquhar, E. A. Feigenbaum, Richard Fikes, Ken Ford, Peter Friedland, Doug Fridsma, Yolanda Gil, Ascuncion Gomez, Tom Gruber, Peter Hall, Peter Hart, Barbara Hayes-Roth, Rick Hayes-

Roth, Carl Hewitt, Torsten Heycke, Donna Holm, Yumi Iwasaki, Jennifer Joseph, Maggie Kimball, Craig Knoblock, Pat Langley, Roger Launius, Henry Lieberman, Henry Lowood, Roberto Mazzolini, the late John McCarthy, Robert E. McGinn, Bill Miller, Andy Neuschatz, H. Penny Nii, Nils Nilsson, Arthur Norberg, Peter Norvig, Pierre Omidyar, Judy O'Neill, Bertram Raphael, Raj Reddy, Bradley Rhodes, Nathan Rosenberg, Stuart Russell, Annalee Saxenian, the late Oliver Selfridge, Ted Shortliffe, the late Herbert A. Simon, Luc Steele, Mark Stefik, Lynn Andrea Stein, Marty Tennenbaum, Shirley Tessler, Mike Travers, Peche Turner, Richard Waldinger, Richard Walker, Robert Wilensky, Terry Winograd, and Lofti Zadeh.

On a personal note, the author would like to thank Lucy Gill, Mary Heldman, Paul Hufstedler, the San Francisco Proust Reading Group, Maria Schopp, Mike Travers, Richard Waldinger, and Jennifer Yeh.

Finally, this book is dedicated to my daughters, Abigail Grace Skinner and Katherine Alice Skinner.

Chapter 16. List of Abbreviations

ACCAT...Advanced Command and Control Architectural Testbed
AAAI.....American Association for Artificial Intelligence
ACM.....Association for Computing Machinery
ACT.....adaptive control of thought
AEC Atomic energy commission
AFIPS.....American Federation of Information Processing Societies
AFOSR.....Air Force Office of Scientific Research
AI.....Artificial Intelligence
ALV.....autonomous land vehicle
ARPA.....Advanced Research Projects Administration
BBN.....Bolt, Beranek, and Newman
CIT.....Carnegie Institute of Technology
CMU.....Carnegie-Mellon University
COSPOP....Committee on Science and Public Policy of the NAS
C&C command and control
C3.....command, communication, and control

C3I.....command, communication, control and intelligence
DARPA.....Defense Advanced Research
Projects Administration
DCA.....Defense Communications Agency
DDR&E director of defense research and engineering;
DEW Distant early warning
DoD.....Department of Defense
GOFAI.....Good Old-Fashioned AI
GPS.....general problem solver
GSIA.....Graduate School of Industrial Administration, at
Carnegie-Mellon University
HLL.....high-level language
HPP.....Heuristic Programming Project
HPCC High Performance Computing and Communications
program
HPS.....Human Problem Solving
IDA.....Institute for Defense Analysis
IEEE.....International Association of Electrical
and Electronic Engineers
IFIP.....International Federation for Information Processing
IJCAI.....International Joint Conference on AI
IPL.....Information Processing Language
IPTO.....Information Processing Techniques Office
IRE.....Institute of Radio Engineers
ISI.....Information Systems Institute, at the
University of Southern California
JSEP.....Joint Services Electronics Program;
KSL.....Knowledge Systems Laboratory, Stanford
LPL.....List Processing Language
LT.....Logic Theorist
MAC.....Man and Computer, multi-access
computer, or Machine-Aided Cognition
MAS.....multi-agent system
MICR.....Magnetic Ink Character Recognition
MIT.....the Massachusetts Institute of Technology
MOD.....Models of Discovery (H.A. Simon)
MOM.....Models of Man (H.A. Simon)
MOML.....Models of my Life (H.A. Simon)
MOBR.....Models of Bounded Rationality (H.A.
Simon)

MOT.....Models of Thought (H.A.Simon)
NACA.....National Advisory Committee for Aeronautics
NASA.....National Aeronautics and Space Council
NAVELEX.....Navy Electronics Systems Command
NIH.....National Institutes of Health
NPL.....National Physical Laboratory, Teddington, England
NRL.....Naval Research Laboratory
(a division of the ONR)
NSS.....Newell, Shaw, and Simon
ONR.....Office of Naval Research
PSS.....physical symbol system
RAND.....Research and development corporation
RFP.....request for proposals
RLE.....Research Laboratory of Electronics, MIT
RTFM..... read the f---- manual
SAC strategic air command
SAGE.....Semi-Automatic Ground Environment
SAIL.....Stanford AI Laboratory
SRI.....originally the Stanford Research Institute, now
the independent SRI International
SRL.....Simulation Research Laboratory
SUMEX-AIM.....Stanford University Medical Experimental Facility,
AI in Medicine
SUR.....speech understanding and recognition
TCP/IP.....Transmission Control Protocol/ Internet Protocol
TIP..... terminal interface message processor
USAF.....United States Air Force

Chapter 17. Bibliography

Abbate, Janet. *Inventing the Internet*. Cambridge, MA: The MIT Press, 2000.

Akera, Atsushi and Nebeker, Frederik, Eds. *From 0 to 1: An authoritative history of modern computing*. New York, NY: Oxford University Press, 2002.

Akera, Atsushi. Calculating a natural world: Scientists, Engineers and Computers during the rise of U.S. Cold War Research. The MIT Press, 2008.

Artificial Intelligence Memoranda: AI Laboratory, MIT 1958-1979. New York: Scientific Datalink, Comtex Scientific 1982. Microfilm.

Banerji, Rana B." The Logic of Learning: A Basis for Pattern Recognition and Improvement of Performance," Advances in Computers 24, 1985: 177-219.

Barr, Cohen, and Feigenbaum. The Handbook of Artificial Intelligence, Volume IV. Menlo Park, CA: Addison-Wesley Publishing Company, 1989.

Bell, C. Gordon." What have we learned from the PDP-11 ?" In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie-Mellon University. New York: Academic Press. 1977: 7-37.

Bell, C. Gordon." What have we learned from the PDP-11 ?" In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie-Mellon University. New York: Academic Press, 1977.

Bonczek, Robert H.; Clyde W. Holsapple, and Andrew B. Whinston." Developments in Decision Support Systems," Advances in Computers 23, 1984: 141-175.

Booth, A.D. Digital Computers in Action. New York: Pergamon Press. 1965.

Bowden." Digital Computation and the Crystallographer", 1953. p203-209.

Brand, Stewart. The Media Lab: Inventing the Future at MIT. New York: Penguin Books, 1988.

Brooks, Ruven." A model of human cognitive behavior in writing code for computer programs, Volume II." CMU. AFOSR TR 75 1085. NTIS paper. May 1975.

Brown, Marc, ed. The Social Responsibility of the Scientist. New York: The Free Press, 1971.

Buchanan, B.G., Sutherland, G.L., and E.A. Feigenbaum." Heuristic Dendral: a program for generating explanatory hypotheses in organic chemistry." In B. Meltzer and D. Michie, eds. Machine Intelligence 4. Edinburgh: Edinburgh University Press. 1969: 209-254.

Buchanan, Bruce, Bobrow, Daniel, Randall Davis, John McDermott, and Edward Shortliffe." Knowledge-Based Systems," Annual Review of Computer Science 1990. 4: 395-416.

Buchanan, B.G. and E.A. Feigenbaum." DENDRAL and Meta-DENDRAL: their applications dimension". Artificial Intelligence 11. 1978: 5-24.

Buchanan, Bruce." Expert Systems: Working Systems and Research Literature". In Stuart Savory, ed. Expert Systems in the Organization. Chichester, U.K.: Ellis Horwood Limited. 1988: 34-74.

Corbato, F.J., interview by Arthur L. Norberg, audio tape(s) and transcript, Cambridge, Mass., 18 April 1989 and 14 November 1990, Charles Babbage Institute Interview. OH 162.

Crevier, Daniel. AI: The Tumultuous History of the Search for Artificial Intelligence. New York: Basic Books, 1993.

Darrach, Brad." Meet Shakey, the first electronic person," Life magazine. November 20, 1970: 58.

Dasgupta, Subrata." Computer Design and Description Languages", Advances in Computers 21 1982: 91-155.

DeJong, Gerald F. "Skimming Newspaper Stories by computer", May 1977. Research Report #104. Yale University Dept CS., ONR research report.

Dickson, Gary W. "Management Information Systems: Evolution and Status", Advances in Computers 20 1981: 1-11.

Dickson, Paul. Think Tanks. New York: Atheneum, 1971.

Djerassi, Carl. The Pill, Pygmy Chimps, and Degas' Horse: The Autobiography of Carl Djerassi. 1992.

Douglas, Deborah. "MIT and War," In Kaiser, David, ed. Becoming MIT: Moments of Decision. MIT Press, 2010: 81-102.

Feigenbaum, Edward, Lederberg and Suppes. "A Study of User Models in intelligent terminal systems". November 1974. Stanford University. From the Edward A. Feigenbaum Papers, SC 240. Box 2, Folder 12. Department of Special Collections and University Archives, Stanford University.

Feigenbaum, Edward. Computer Sci Dept, Stanford U. "Supplemental Proposal to DARPA. Computer Based Consultant for Structural Design Problem Solving Services" June 1977. From the Edward A. Feigenbaum Papers, SC 240. Department of Special Collections and University Archives, Stanford University.

Fennell, R.D., and V. Lesser. "Parallelism in AI problem solving: a case study of Hearsay II", CMU. 1973. Report AFOSR TR 76 6598.

Fikes, Richard A., et al. "Learning and Executing Generalized Robot Plans." September, 1972. SRI. Army Research Office- Durham AD 765 242.

Freiberger, Paul, and Swaine, Michael. Fire in the Valley: the making of the personal computer. Berkeley, CA: Osborne/ McGraw-Hill. 1984.

Gammill, Robert C. "Microcomputers: Applications, Problems, and Promise", Advances in Computers 21 1982: 155-224.

Garvin, Paul. "The Current State of Language Data Processing", Advances in Computers 24, 1985: 217-275.

Gerhardt, Douglas, et al. "Window: A Formally Specified Graphics Based Text Editor" AFOSR TR 73 1131.

Gillogly, James J. "The technology chess program". CMU. Dept of CS. 29 November 1971. CMU CS 71 109.

Goldstein and S. Papert. "Artificial Intelligence, Language, and the Study of Knowledge" Cognitive Science. 1977.

Grape, Gunnar Rutger. "Model-Based [Intermediate-level] computer vision," CS Department, Stanford University. May, 1973.

Grosch, Herbert R. Computer: bit slices from a life. Novato, California: Underwood-Miller, 1991.

Hafner, Katie and Matthew Lyon. Where Wizards Stay Up Late: The Origins of the Internet. New York: Touchstone/ Simon & Schuster, 1998.

Hall, Mark, and John Barry. Sunburst: The Ascent of Sun Microsystems. Chicago, IL: Contemporary Books, Inc. 1990.

Hapgood, Fred. "Computers Aren't so Smart After All" The Atlantic 234 (August 1974): 37-45.

Hapgood, Fred. Up the infinite corridor: MIT and the technical imagination. Reading, MA: Addison-Wesley. 1993.

Heath, D. "Large scale integration in electronics". Scientific American 222 (February 1970): 22-31. Wheaton reader (Evolution of modern technology, U.C. Berkeley).

Hiltzik, Michael. Dealers of lightning. Xerox Parc and the dawn of the computer Age. New York: Collins Business, 1999.

Hart, Peter E.; Nils Nilsson; and Ann E. Robinson. "A Causality Representation for Enriched Robot Task Domains." SRI. ONR Report on SRI Project 1187. December 1971.

Hayes-Roth, Frederick, Donald Waterman, and Douglas B. Lenat. Building Expert Systems. Reading, MA: Addison-Wesley. 1983.

Hilts, Philip J. Scientific temperaments: Three Lives in Contemporary Science. New York: Simon and Schuster. 1982.

Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the Computer Science department, Carnegie Mellon University. New York: Academic Press. 1977: 183-198.

Kaiser, David, ed. Becoming MIT: Moments of Decision. MIT Press, 2010.

Kaiser, David. "Elephant on the Charles: Postwar Growing Pains", in Kaiser, David, ed. Becoming MIT: Moments of Decision. MIT Press, 2010: 103-121.

Klahr, David, Pat Langley, and Robert Neches, eds. Production system models of learning and development. Cambridge, MA: The MIT Press, 1987.

Klahr, Philip, and Donald Waterman, "Artificial Intelligence: A RAND Perspective", The AI Magazine, Summer 1986, 56-64.

Klahr, David and Kenneth Kotovsky, eds. Complex information processing: the impact of Herbert A. Simon. Hillsdale, N.J.: L.L. Erlbaum Associates. 1989.

LeCuyer, Christophe. Making Silicon Valley Innovation and the growth of high tech, 1930-1970. The MIT Press, 2006.

Leslie, Stuart. "How the West was Won: The Military and the Making of Silicon Valley" In William Aspray, ed., Technological Competitiveness: Contemporary and Historical Perspectives on the

Electrical, Electronics and Computer Industries. New York: IEEE Press. 1993: 75-89.

Leslie, Stuart. *The Cold War and American Science: The Military-Industrial-Academic Complex at MIT and Stanford*, 1993.

Leslie, Stuart. " 'Time of Troubles' for the Special Laboratories", In Kaiser, David, ed. *Becoming MIT: Moments of Decision*. MIT Press, 2010: p124-143.

Levy, Steven. *Hackers: Heroes of the Computer Revolution*. Garden City, N.Y.: Anchor Press-Doubleday. 1984. Delta Trade Paperbacks (Bantam Doubleday). 1994.

Kidder, Tracy. *The Soul of a New Machine*. Boston: Little, Brown, & Co. 1981.

Licklider, J.C.R. " The Early Years: Founding IPTO". In Bartee, Thomas, ed. Expert Systems and Artificial Intelligence: Applications and Management. Philadelphia: Howard W. Sams & Company (MacMillan). 1988: 219-227.

Licklider, J.C.R. " Man-Computer Symbiosis". IRE Transactions on Human Factors in Electronics. (March 1960): 4-11. (n.b.: IRE is now IEEE).

Lindsay, Robert K., Buchanan, B.G., E.A. Feigenbaum, and Joshua Lederberg. " DENDRAL: a Case Study of the first expert system for scientific hypothesis formation". Artificial Intelligence 61 (1993): 209-261.

Lowood, Henry. " From Steeples of Excellence to Silicon Valley: The Story of Varian Associates and Stanford Industrial Park". Stanford University Archives.

McCarthy, epistemological problems in AI; in readings in knowledge representation; by Brachman and Levesque; also is in IJCAI 1977.

McCarthy, John." First order theories of individual concepts and propositions," 1979. Reprinted in Readings in knowledge representation; by Brachman and Levesque.
also Naive physics, see Hayes articles in same book.

McCarthy, J." Programs with Common Sense." In Luger, George, ed. Computation and Intelligence: Collected Readings. Cambridge, MA: The MIT Press. 1995.

McCarthy, John." Reminiscences on the History of Time Sharing". <http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html>. 1983.

McCarthy, John." Memorandum to P.M. Morse proposing Time Sharing". <http://www-formal.stanford.edu/jmc/history/timesharing-memo/timesharing-memo.html>. January 1, 1959.

McCorduck, Pamela." An Introduction to the Fifth Generation". Communications of the ACM. 26, 9. September, 1983: 629-630.

McCorduck, Pamela. Machines Who Think, A Personal Inquiry into the History and Prospects of Artificial Intelligence. San Francisco, CA: W.H. Freeman and Company. 1979.

McDermott, D.; Newell, A.; and J. Moore." The Efficiency of Certain Production System Implementations," September 1976. NTIS document, AD A034 851.

McDermott, J." R1: An expert in the computer systems domain". In AAAI 1. 1980: 269-271.

McDermott, J." R1: The formative years". AI Magazine. 2,2. 1981: 21-29.

McDermott, J." R1: a rule-based configurer of computer system". Artificial Intelligence. 19, 1. September, 1982: 39-88.

McKenney, James L, with Copeland, Duncan G., and Richard O. Mason. Waves of Change: Business Evolution through Information Technology. Boston, MA: Harvard Business School Press. 1995.

Minsky, Marvin." A Selected Descriptor-Indexed Bibliography to the Literature on Artificial Intelligence". In Feigenbaum, Edward A. and Julian Feldman, eds. Computers and Thought. New York: McGraw-Hill. 1963: 453-524.

Minsky, Marvin.' Frames', in Patrick Winston ed., The Psychology of Computer Vision. 1975.

Minsky, Marvin." Steps toward Artificial Intelligence" in Computers and Thought. 1963.

Minsky and Papert." 1968-1969 Progress Report." MIT AI Memo 200.

Morgan, Jane. Electronics in the West: The First Fifty Years. Palo Alto; National Press Books. 1967.

Neisser, Ulric." The imitation of man by machine", Science #139, 1963: 193-197.

Nelkin, Dorothy. The University and Military Research: Moral Politics at MIT. Ithaca, N.Y.: Cornell University Press. 1972.

Newell, A. and G. Robertson." C.mmp: A Progress Report on Synergistic Research." In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie Mellon University. New York: Academic Press. 1977.

Newell, A." The chess machine: an example of dealing with a complex task by Adaptation". Proceedings of the Western Joint Computer Conference (7, 1955: 101-108).

Newell, Allen." Some Problems of Basic Organization in Problem-Solving Programs". In Yovits, Marshall C., G.T. Jacobi, and G.D. Goldstein, eds. Conference on Self-Organizing Systems. Washington, D.C.: Spartan Books. 1962: 93-423.

Newell, A. Learning Generality, and problem solving. Amsterdam: North Holland. 1963. Proceedings of the IFIP Congress 62 407-412.

Newell, Allen. "Heuristic programming: Ill-structured problems". In J. Aronofsky, ed. Progress in Operations Research. New York: Wiley. 1969: 360-414.

Newell, A. "You can't play Twenty Questions with Nature and Win: Projective comments on the papers of this Symposium." In Chase, William G., ed. Visual Information Processing. Proceedings of the 8th Symposium on Cognition. 1972, Carnegie-Mellon University. New York, Academic Press, 1973.

Newell, Allen. "On the analysis of human problem-solving protocols". In Johnson-Laird and Wason, eds. Thinking: Readings in Cognitive Science. Cambridge, England: Cambridge University Press. 1977: 46-60.

Newell, Allen. "The Knowledge Level". Presidential Address, American Association for Artificial Intelligence, AAAI80, Stanford University 19 August 1980. The AI Magazine 2, 2 (Summer, 1981): 1-20.

Newell, Allen. Cognitive Science 1980 vIV, 135-183.

Newell, Allen. "Reflections on the Structure of an Interdiscipline," In Machlup and Mansfield, eds. The Study of Information: Interdisciplinary Messages. New York: John Wiley & Sons. 1982: 99-109.

Newell, Allen. "Intellectual Issues in the History of AI," In Machlup, and Mansfield, eds. The Study of Information: Interdisciplinary Messages. New York: John Wiley & Sons. 1982: 185-210.

Newell, Allen. "Reports on Artificial Intelligence from Carnegie-Mellon University: Introduction to the COMTEX Microfiche Edition," in Robert Englemore, editor. Readings from AI magazine: volumes 1-5. 1988: 328-332.

Newell, Allen." The Serial Imperative". In Baumgartner, Peter, and Payr, Sabine, eds. Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists. Princeton, NJ: Princeton University Press. 1995: 145-156.

Newell, A., Shaw, J.C. and Simon, H.A." Report on a General Problem-Solving Program". Proceedings of the International Conference on Information Processing.(ICIP). Paris: UNESCO House. 1959: 256-264.

Newell, A.M. and H. Simon." GPS, A Program that Simulates Human Thought". Lernende Automaten, Munich: R. Oldenbourg KG. 1961. Reprinted in Feigenbaum, Edward A. and Julian Feldman, eds. Computers and Thought. New York: McGraw-Hill. 1963: 279-297.

Newell, Allen and Herbert A. Simon." Computer Science as Empirical Inquiry: Symbols and Search." Communications of the ACM 19, 3 (1976). Also in J. Haugeland, ed., Mind Design: Philosophy, Psychology, Artificial Intelligence. Cambridge, MA: The MIT Press. 1980: 35-66.

Newell, Allen and H.A. Simon." The Process of Creative Thinking," In Howard E. Gruber, Glenn Terrell, and Michael Wertheimer. Contemporary Approaches to Creative Thinking. New York: Atherton Press. 1962: 63-119.

Newell and Simon, Human Problem Solving. 1972.

Newell, Allen." Heuristic programming: Ill-structured problems". In J. Aronofsky, ed. Progress in Operations Research. New York: Wiley. 1969: 360-414.

Newell, Allen." The Serial Imperative". In Baumgartner, Peter, and Payr, Sabine, eds. Speaking Minds: Interviews with Twenty Eminent Cognitive Scientists. Princeton, NJ: Princeton University Press. 1995: 145-156.

Newell, Allen and C.G. Bell- Computer Structures- Readings and Examples; 1971.

Newell, Allen." The Mind's eye in chess"; Chase, ed. visual information processing. 1973.

Newell, A., Shaw, J.C. and Simon, H.A." Report on a General Problem-Solving Program". Proceedings of the International Conference on Information Processing.(ICIP). Paris: UNESCO House. 1959: 256-264.

Newell, A.M. and H. Simon." GPS, A Program that Simulates Human Thought". Lernende Automaten, Munich: R. Oldenbourg KG. 1961. Reprinted in Feigenbaum, Edward A. and Julian Feldman, eds. Computers and Thought. New York: McGraw-Hill. 1963: 279-297.

Newell, A. Learning Generality, and problem solving. Amsterdam: North Holland. 1963. Proceedings of the IFIP Congress 62 407-412.

Newell, Allen and Herbert A. Simon." Computer Science as Empirical Inquiry: Symbols and Search." Communications of the ACM 19, 3 (1976). Also in J. Haugeland, ed., Mind Design: Philosophy, Psychology, Artificial Intelligence. Cambridge, MA: The MIT Press. 1980: 35-66.

Newell, A." The chess machine: an example of dealing with a complex task by Adaptation". Proceedings of the Western Joint Computer Conference (7, 1955: 101-108).

Nilsson, Nils." SRI Artificial Intelligence Center Technical Notes: an Introduction to the COMTEX Microfiche Edition," in R Engelmores, editor. Readings from AI magazine: volumes 1-5. Originally published Spring 1984. 1988: 318-327.

Nilsson 1988 Charles Babbage Institute Oral History interview.

Nilsson NTIS report 1973. AD 761 641.

Nilsson, Nils. The quest for artificial intelligence : A history of ideas and achievements. Cambridge University Press, 2010.

Web version: <http://ai.stanford.edu/~nilsson/QAI/qai.pdf>

Norberg, Arthur L. "The Origins of the Electronics Industry on the Pacific Coast", 1976. Proceedings of the IEEE. 64 (1976): 1314-1322.

Norberg, Arthur L., and Judy O'Neill. Transforming Computer Technology: Information Processing for the Pentagon 1962-1986. Johns Hopkins Studies in the History of Technology. 2000.

O'Mara, Margaret Pugh. Cities of Knowledge: Cold War Science and the Search for the Next Silicon Valley. Princeton University Press. 2005.

O'Neill, Judy. doctoral dissertation. history of timesharing. The University of Minnesota, 1992.

Penick, James L. Jr.; Carroll W. Pursell, Jr.; Morgan B. Sherwood; and Donald C. Swain, eds. The Politics of American Science, 1939 to the Present. Revised edition. Cambridge, MIT Press. 1972.

Peterson, T.F. Nightwork: A history of hacks and pranks at MIT. The MIT Press. 2003.

Pylyshyn, Zenon, ed. The Robot's Dilemma: the Frame Problem in Artificial Intelligence. Norwood, N.J., Ablex. 1987.

Press, Larry. "Before the Altair: the History of Personal Computing", C-ACM 36, 9 (September, 1993): 27-33.

Reddy, R., and A Newell. "Multiplicative Speedup of Systems," In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie Mellon University. New York: Academic Press. 1977: 183-198.

Rheingold, Howard. Tools for Thought: The People and Ideas behind the Next Computer Revolution. 1985.

Riesbeck, Christopher, and Schank, Roger. "Comprehension by Computer: Expectation-Based Analysis of Sentences in Context." Yale University, Research Report #78. October, 1976. (NTIS paper).

Salisbury, David and Gio Wiederhold. <http://www-db.stanford.edu/pub/voy/museum/Forsythenews.html> George Forsythe his vision and its effects; David Salisbury, Stanford News Service. "Note in response to a query". 11 26 1997.

Schank, Roger and Charles Rieger III." Inference and the computer understanding of natural language," in readings in knowledge representation; by Brachman and Levesque; 1974.

Shurkin, Joel. Broken genius: The rise and fall of William Shockley Creator of the electronic age. New York: MacMillan, 2006.

Simon, H.A. Models of Thought. New Haven, CT: Yale University Press. 1979.

Simon, Herbert. Models of man: social and rational; mathematical essays on rational human behavior in a social setting. New York: Wiley. 1957.

Simon, Herbert A. and Laurent Siklossy, eds. Representation and meaning; experiments with information processing systems. Englewood Cliffs, N.J., Prentice-Hall. 1972.

Simon, H.A." Scientific Discovery and the Psychology of Problem Solving". Models of Discovery. Boston, MA: D. Reidel. 1977. (Originally pub. 1966).

Simon, H.A., with E.A. Feigenbaum." A Theory of the Serial Position Effect: Effects of similarity, familiarization and meaningfulness in verbal learning". In MOT (orig. 1964): 114-123.

Simon, H.A., with Lee W. Gregg." One trial and incremental learning" in MOT (1967). 124-132.

Simon, H.A." A Behavioral Model of Rational Choice". In MOT. Orig 1955. 7-19.

Simon, H.A." Rational Choice and the Structure of the Environment". In MOT. Orig. 1956: 20-28.

Simon, H.A." Motivational and Emotional Control of Cognition". MOT. Orig. 1967: 29-38.

Simon, H.A." How Big is a Chunk ?" in MOT. Orig. 1974: 50-61.

Simon, H.A., with Allen Newell and J.C. Shaw." The process of Creative Thinking" in MOT. Orig. 1962.: 144-174.

Simon, H.A., with Jeffrey M. Paige." Cognitive Processes in Solving Algebra Word Problems" In MOT. Orig. 1966. 201-229.

Simon, H.A., with Kenneth Kotovsky." Human Acquisition of Concepts for Sequential Patterns." In MOT. Orig. 1963: 263-273.

Simon, H.A." On the Nature of Understanding," In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie Mellon University. New York: Academic Press. 1977.

Simon, Herbert. Models of man: social and rational; mathematical essays on rational human behavior in a social setting. New York: Wiley, 1957.

Simon, Herbert A. and Laurent Siklossy, eds. Representation and meaning; experiments with information processing systems. Englewood Cliffs, N.J., Prentice-Hall. 1972.

Simon, H.A." Scientific Discovery and the Psychology of Problem Solving". Models of Discovery. Boston, MA: D. Reidel. 1977. (Originally pub. 1966).

Simon, H.A., with E.A. Feigenbaum." A Theory of the Serial Position Effect: Effects of similarity, familiarization and meaningfulness in verbal learning". In MOT (orig. 1964): 114-123.

Simon, H.A., with Lee W. Gregg." One trial and incremental learning" in MOT (1967). 124-132.

Simon, H.A." A Behavioral Model of Rational Choice". In MOT. Orig 1955. 7-19.

Simon, H.A. "Rational Choice and the Structure of the Environment". In MOT. Orig. 1956: 20-28.

Simon, H.A. "Motivational and Emotional Control of Cognition". MOT. Orig. 1967: 29-38.

Simon, H.A. "How Big is a Chunk?" in MOT. Orig. 1974: 50-61.

Simon, H.A., with Allen Newell and J.C. Shaw. "The process of Creative Thinking" in MOT. Orig. 1962: 144-174.

Simon, H.A., with Jeffrey M. Paige. "Cognitive Processes in Solving Algebra Word Problems" In MOT. Orig. 1966. 201-229.

Simon, H.A., with Kenneth Kotovsky. "Human Acquisition of Concepts for Sequential Patterns." In MOT. Orig. 1963: 263-273.

Simon, H.A. "On the Nature of Understanding," In Jones, Anita K., ed. Perspectives on Computer Science: from the tenth anniversary symposium at the CS department, Carnegie Mellon University. New York: Academic Press. 1977: 199-215.

Snyder, Lawrence. "Supercomputers and VLSI: The Effect of large-scale integration on Computer Architecture", Advances in Computers 23, 1984: 1-33.

de Solla Price, Derek J. Little Science, Big Science, and Beyond. New York: Columbia University Press, 1963.

Sridharan, N.S. "Search strategies for the task of organic chemical synthesis," Stanford AI project. AIM 217. August 1973. Text is also in IJCAI.

Stanford Knowledge Systems Laboratory's Applied Artificial Intelligence Research: The Heuristic Programming Project, 1990.

Stork, David, editor. HAL's Legacy: 2001's Computer as Dream and Reality. Cambridge, MA: The MIT Press, 1997.

Stratton, Julius A., and Loretta H. Mannix. Mind and Hand: The Birth of MIT. The MIT Press. 2005.

Sutherland, Ivan." Computer Displays," Scientific American (June 1970): p56-70.

Walker, Richard." Another round of globalization in San Francisco". Urban Geography 1996: P60-94.

Weyhrauch, Richard W." Prolegomena to a theory of mechanized formal reasoning," Artificial Intelligence 12 (1,2, 1980): 133-170. Reprinted in Readings in knowledge representation; by Brachman and Levesque.

Wiener, Norbert. The Human Use of Human Beings: Cybernetics and Society. Houghton Mifflin Company: Boston, 1950.

Wildes, Karl and Nilo Lundgren. A Century of Electrical Engineering and Computer Science at MIT 1882-1982. The MIT Press. 1985.

Winston, Patrick, ed. The Psychology of Computer Vision. 1975.

Winston, Patrick." 'Arch-Learner', in Winston ed., The Psychology of Computer Vision. 1975.

Winston, Patrick and Karen A. Prendergast, Eds. The AI Business: Commercial uses of Artificial Intelligence. The MIT Press.

Wolfe, Tom." The Tinkerings of Robert Noyce: How the Sun Rose on the Silicon Valley," Esquire Magazine, December, 1983: 346-374.

Zachary, G. Pascal. Endless Frontier: Vannevar Bush Engineer of the American Century. Cambridge, MA: The MIT Press, 1999.

Chapter 18. Endnotes

- (1) Jane Jacobs.
- (2) *Silent Spring* Rachel Carson, 1963.
- (3) This is not intended as a rationalization of the Cold War or of warfare per se. The aphorism says it all: "War is good business, invest your sons."
- (4) Kenneth Flamm, *Creating the Computer*. Also see Jon Ajar's The Government Machine, and works in the competitiveness discussions, notably those of Chalmers Johnson and Marie Anchoy.
- (5) *Computers and Thought*.
- (6) Samuel 1959- arthur program
- (7) Slagle, calculus program;
- (8) "Atrial and error search." (Minsky 1968, p12).
- (9) Minsky 1968, p7.
- (10) "Despite its contemporary association with science fiction, the term "cyborg" actually originated in science. Manfred Clynes and Nathan S. Cline coined the word in 1960 to describe human machine systems for space exploration; in their definition the cyborg 'deliberately incorporates exogenous components extending the self-regulatory control [f] of the organism in order to adapt it to new environments (Manfred Clynes and Nathan S Kline, " Cyborgs and Space," Astronautics September 1960).' Paul Edwards, *The Closed World*, 1996, p325.
- (11) Marvin L. Minsky, Charles Babbage Institute OH Interview #179, 1989.
- (12) Campbell-Kelly and Aspray 1996, p130.
- (13) Campbell-Kelly and Aspray 1996, p130
- (14) Reid 1984, p61.
- (15) Misa 1984.

Also:

Bell Telephone Laboratories Technical Staff. A History of Engineering and Science in the Bell System. National Service in War & Peace, 1925-1975. Bell Telephone Laboratories. 1975.

Bell Telephone Laboratories Technical Staff. A History of Engineering and Science in the Bell System. The Early Years (1875-1925). Bell Telephone Laboratories. 1975.

- (16) Reid 1984, p13-15.
- (17) Williams 1979 p392.

(18) Williams 1979 p392; picture of the 7090, lots and lots of cabinets.

(19) Martin, C. Dianne. "The Myth of the Awesome Thinking Machine", Communications of the ACM 36, 4 (April, 1993): 120-133.

(20) Paul Armer, "Attitudes Toward Intelligent Machines". In Feigenbaum, Edward A. and Julian Feldman, eds. Computers and Thought. New York: McGraw-Hill. 1963: 389-405. Reprinted in Pylyshyn, Zenon, ed. Perspectives on the Computer Revolution. Englewood Cliffs, N.J. 1970.

(21) Armer 1963, p394.

Armer cites a programming guide for the latter bit of wisdom: Andree R.V. 1958. Programming the IBM 650 Magnetic Drum Computer and Data processing machine. New York. Holt.

(22) Taube 1961, p9. Taube's book, inspired by a Science article by Norbert Wiener, seems to have been rather widely read, but is simply a poor and poorly written compendium of most of the anti-computing arguments taken together.

(23) Turing 1950, p20.

(24) Armer 1963, p393.

(25) *ibid.* p394; Turing 1950, p27.

(26) Wiener used a new version of the slavery-dumb machines argument, asserting that in the nuclear age this was something more frightening than reassuring. He proffers that intelligent computing programs are a moral problem as well as a security risk:

"The problem, and it is a moral problem with which we are here faced is very close to one of the great problems of slavery. Let us grant that slavery is bad because it is cruel. It is however self-contradictory and for a reason which is quite different. We wish a slave to be intelligent, to be able to assist us in the carrying out of our tasks. However, we also wish him to be subservient. Complete subservience and complete intelligence do not go together. ...if the machines become more and more efficient and operate at a higher and higher psychological level, the catastrophe foreseen by Butler of the dominance of the machines comes nearer and nearer." (Wiener 1960, p1357).

Norbert Wiener; Science May 6, 1960: "Some moral and technical consequences of automation"; 1355-1359.

(27) Turing points out that these ways of thinking are "mostly founded on the principle of scientific induction. A man has seen thousands of machines in his lifetime. From what he sees of them he draws a number of general conclusions. They are ugly, each is designed for a very limited purpose..." (Turing 1950, p24).

It is interesting to note that Armer defends computer memory potential on the basis of the promise of thin film memory rather because of integrated circuitry (Armer 1963, p395).

(28) Dianne C. Martin, "The Myth of the Awesome Thinking Machine", Communications of the ACM 36, 4 (April, 1993): 120-133.

(29) Ironic as it is to observe, the most pacifistic countries at this time were Japan and Germany, forbidden from maintaining standing armies, and instead developing at breakneck speed as industrial powers.

(30) Personal interview, July 1994, Stanford Medical School Professor Ted Shortliffe.

(31) Shurkin 1984 and Reid 1984; also Noyce.

(32) Reid 1984, p23.

(33) Mandl 1967, p197.

(34) Noyce 1977, p64.

(35) Reid 1984, p116.

(36) Reid 1984, p116; Wolfe 1983, p121-123.

(37) Reid 1984, p121

(38) *ibid.*, p123

(39) Wolfe 1983, p350

(40) 1977, p64.

(41) Dasgupta p30.

(42) Semiconductors, despite themselves, were at least initially just as infuriating in their microscopic embroidery as transistors had been:

"The work bays, where the transistors were produced looked like slightly sunnier versions of the garment sweatshops of San Francisco's Chinatown. Here were rows of women hunched over worktables, squinting through microscopes doing the most tedious and frustrating sort of manual labor, cutting layers of silicon apart with diamond cutters, picking little rectangles of them up with tweezers, trying to attach wires to them, dropping them, rummaging around on the floor to find them again, swearing, muttering, climbing back up to their chairs, rubbing their eyes, squinting back through the microscopes, and driving themselves crazy some more. Depending on how well the silicon or germanium had been cooked and doped, anywhere from 50 to 90 percent of the transistors would turn out to be defective even after all that, and sometimes the good ones would be the ones that fell on the floor and got ruined." (Wolfe 1983, p350).

(43) Wolfe 1983, p350. "They had decided to move into the most backward area of computer technology, which was data storage, or "memory." A computer's memory was stored in ceramic ringlets known as cores. Each ringlet contained one "bit" of information, a "yes" or a "no," in the logic of the binary system of mathematics that computers employ. Within two years Noyce and Moore had developed the 1103 memory chip, a chip of silicon and polysilicon the size of two letters in a line of type. Each chip contained four thousand

transistors, did the work of a thousand ceramic ringlets, and did it faster. The production line still consisted of rows of women sitting at tables as in the old shed-and-rafter days, but the work bays now looked like something from out of an intergalactic adventure movie. The women engraved the circuits on the silicon photographically, wearing antiseptic Mars Voyage suits, headgear, and gloves because a single speck of dust could ruin one of the miniature circuits. The circuits were so small that "miniature" no longer sounded small enough. The new word was "microminiature." Everything now took place in an air- conditioned ice cube of vinyl tiles, stainless steel, fluorescent lighting, and backlit plastic." (Wolfe 1983, p350).

Also re Intel: Tim Jackson, Inside Intel: Andy Grove and the World's Most Powerful Chip Company. 1997.

(44) The source for most of the materials above regarding NASA, NACA, Sputnik and its brethren is the Aeronautics and Astronautics Chronology, compiled by NASA Historian Eugene M. Emme, Emme, at nasa's gov website office/ pao history timeline. McDougall's The Heavens and the Earth also provides a detailed narrative.

(45) February 7 58: The Advanced Research Projects Agency (ARPA) was established by the DOD, and Roy W. Johnson, a vice president of General Electric Co., was appointed by Secretary of Defense McElroy as its Director. ARPA was placed in charge of the Nation's outer space program.

(46) See James R. Killian, Jr., Sputnik, Scientists, and Eisenhower: A Memoir of the First Special Assistant to the President for Science and Technology. Cambridge, MA: The MIT Press. 1977.

(47) Dickson 1971, p11.

(48) Licklider 1988, p219; Licklider, CBI OH Interview 150, 1988.

(49) CBI OH 163, 1989.

(50) Reid 1984, p116.

(51) The Apollo program: U.S. space exploration program to land men on the moon, was initiated in May 1961 by President JFK. It achieved its objective on July 20, 1969, when Neil Armstrong set foot on the Moon. The program terminated with the successful Apollo-Soyuz linkup in space during July 1975. It placed more than 30 astronauts in space and 12 on the Moon. Op cit. NASA Chronology.

(52) Reid 1984, p116.

(53) Reid 1984, p121-123.

(54) Norberg and O'Neill 1996, p9

(55) Norberg and O'Neill 1996, p10.

We add a note regarding the IDA because its path crossed with AI at various points. This is particularly the case because of its advisory role in oversight over AI.

" The Institute for Defense Analysis was formed in 1956 at the request of the Secretary of Defense Charles E. Wilson in an effort to attract and retain a permanent corps of civilians to aid in the Weapons Systems Evaluation Group of the Joint Chiefs of Staff. The evaluation group needed help in arbitrating the costly, earthshaking missile and weapons battles which brewed during Defense expansion in the 1950s (Dickson 1971, p146).

... Wilson asked MIT to form such a group, and it agreed to do so but only if other universities helped to form the corporation. CalTech, the Case Institute of Technology, Stanford, Tulane, and MIT acted as co founders and were later joined by U.C., the University of Chicago, Columbia, the University of Illinois, University of Michigan, Penn State and Princeton. A \$500 thousand Ford Foundation grant got it into operation... As the years progressed, IDA's spheres of influence broadened. It became advisor to the Sec of Defense himself on the two most important elements of the Pentagon's technical programs: the ARPA and the Office of Defense Research and Engineering. By the late 1960s it had grown to a corporation with five divisions, an annual budget of about \$14 million and ...600 employees." (Dickson 1971, p146).

(56) IPTO has had other names. IPTO was renamed the Information Science and Technology Office, ISTO, in 1986. ISTO was reorganized in May 1991 into the Computing Systems Technology Office (CSTO) and the Software and Intelligent Systems Technology Office (SISTO). ARPA's own name changed too: the "Defense" prefix was added in 1971, turning ARPA into DARPA. Then in 1993, the name was changed back to ARPA. We will refer to either ARPA or DARPA, as chronologically appropriate. Norberg and O'Neill 1992, p.xv.

(57) Now we know what Uncle Sam wanted from the AI people: blue-sky research which might, if only tangentially, have some application to military objectives. We might also ask: What did Uncle Joe want ? Notwithstanding that Stalin himself had died in 1953, his evil spirit lived on. Soviet systems computing was functional and competent, if limited to the military elite and kept from intellectuals and dissidents. See Nemes, Arkadev, Armer article in C&T; also Berenyi, Ivan." Computers in Eastern Europe," Scientific American (October, 1970): 102-109. Also E. Feigenbaum; Soviet cybernetics and computer sciences; C ACM Dec 1961; 566-579; re his site visit to the USSR.

(58) Licklider 1988, p219.

(59) Fleck 1982; Rheingold 1985; Norberg and O'Neill 1992; Bartee 1988, Engelmores 1988.

(60) Norberg and O'Neill 1992, p121.

(61) Norberg and O'Neill 1992, p121.

(62) The usual critique of the military-industrial complex rests in part on the basis of cost overruns and alleged inefficiencies in the way that military 'products' are produced; see, e.g., Seymour Melman's scathing diatribes against the military economy. The IPTO greatly deviated from this pattern of operation.

(63) Licklider 1988, p220-221.

(64) Bartee, 1988.

(65) Licklider, CBI OH Interview 150, 1988.

(66) Ivan Sutherland, CBI OH interview 171, 1989.

(67) Michael L. Dertouzos of MIT discusses this issue in CBI OH 164, 1989. Ivan Sutherland did go into business with Evans when he was at the University of Utah, and Douglas Engelbart and Marvin Minsky have been associated with commercial enterprises, although these have been very much at the periphery of their work.

(68) Newell and Simon 1972, p.viii.

(69) Regarding the AFOSR, see Saul Amarel CBI interview.

(70) EAF 1989, CBI #157.

(71) EAF CBI interview 1989 #157.

(72) CBI OH interview.

(73) Hafner.

(74) Lawrence G. Roberts, CBI OH Interview 159, 1989.

(75) Ivan Sutherland, CBI OH interview 171, 1989.

(76) Licklider 1965, p90.

(77) Licklider 1960, p306.

(78) Licklider 1960, p309-310.

(79) *ibid.*, p315.

(80) The MIT community researchers who JCR Licklider enumerated as participating in his research group were John McCarthy, Marvin Minsky, computational linguist Bertold Bloom, Daniel Bobrow, Richard Kahn, David Park, and Bertram Raphael, then at MIT.

(81) " Shortly after the text was written, "bulk core" memories with 18 million bits per unit, and as many as four units per computer, were announced for delivery in 1966." Licklider 1965, p17.

(82) Licklider 1965, p129.

(83) Licklider 1960, p316.

(84) J.C.R. Licklider, CBI OH Interview 150, 1988

(85) Licklider 1960, p308, p316.

(86) Licklider 1960, p308

(87) At least Fredkin is not mentioned in Englebart oral history; see also Parsaye, p229. Englebart had, however, read the Bush paper when it appeared in The Atlantic Monthly in 1945.

(88) Campbell-Kelly and Aspray 1996, p130.

(89) Mandl 1967, p172.

(90) Mandl 1967, p170.

(91) Mandl 1967, p180.

(92) Booth 1965.

(93) Campbell-Kelly, Aspray 1998.

(94) Booth 1965, p30.

(95) Wegner ref #27; Leathrum p4, and Backus et al. 1957.

(96) Campbell-Kelly and Aspray 1996, p192; Wegner 1976, p7.

(97) Booth 1965, p21; Mandl 1967, p235 has a photo

(98) Williams 1979, p393.

(99) Mandl 1967, p210; also RES can mention people who said that they used punched cards in the 70s and 80s.

(100) Libraries of the Future.

(101) Mandl 1967, p228.

(102) These are called electric typewriters, but actually were much more complex and resembled them only in the keyboard: "These are used as offline equipment for paper tape preparation manually. On line operation could be performed, though the comparatively slow write process compared w the computers internal speed of operation limits such use... For on line operation, encoder matrix systems are employed to produce a binary output in parallel form [then diagram, interesting] . The diode matrix binary encoder shown has a pulse source which is triggered to produce a single pulse OP such as is procured from a single shot multivibrator. If for instance the 5 key is depressed the pulse generated would pass through the diodes and enter the first and third vertical lines from the right... producing the parallel binary OP # 0101..." Mandl 1967, p218.

(103) Pfeiffer 1962, p1, with photo.

(104) " For expediting the training of computer personnel, a number of manufacturers have marketed computer trainers, which are in essence, miniature digital computers containing logic circuits, flip flop counters, and registers, as well as useful.. storage capabilities. Such trainers... serve as a convenient demo item in

schools having computer courses. The design features... permit students to be exposed to hardware assembly practices, interconnection of logic components and the general philosophy of programming techniques. Some of these trainers have avail peripheral equipment for read in and read out purposes and for facilitating programming procedures." (Mandl 1967, 332).

(105) Levy 1984.

(106) Booth, A.D. Digital Computers in Action. New York: Pergamon Press. 1965. Booth 1965, p21.

Alternatively, Mandl's tone is scientific but cheerful:
" Direct or indirect methods can be used for IP/OP communication w the computer. For direct input, the instructions and data to be processed can be applied with an electric typewriter or pushbutton controls. Since this is a slow procedure compared with the computer's inherently rapid processes, such piecemeal data insertion is only used on special occasions when it is necessary to correct program errors (debugging) or to test programs" (Mandl 1967, p210).

(107) McCarthy 1983.

(108) Levy 1984, p26-28.

When the TX-0 arrived at MIT, it had been stripped down since its days at Lincoln Laboratory; the memory had been reduced considerably, to 4096 words of 18 bits each; and came w no software; so Jack Dennis had started writing systems programs; first, Dennis worked on an assembler; i.e. to turn hacker (assembly) code into object code, or machine code; also provided a debugger; circa this time, the idea of a "handle" was invented. Levy 1984, p36.

This history is also discussed in Norberg and O'Neill 1992, p314.

Regarding the connection between the Lincoln Laboratory and DEC, see C. Gordon Bell, J Craigmudge and John E McNamara, Computer Engineering: A DEC View of Hardware System Design. Bedford, MA: Digital Press. 1978.

(109) Levy 1984, p26-28

(110) Shurkin 1984, p305-306.

(111) Levy 1984, p50

(112) Levy 1984, p74

(113) See Hafner and Lyon, Where Wizards Stay up late: the origins of the Internet, 1996, for a thorough account.

The magisterial overview in DARPA Technical Accomplishments, by R.H. Van Atta, S.J. Deitchman, and S.J. Reed (IDA Paper P-2192. Washington, D.C.: IDA (Institute for Defense Analyses). 1991,

contains a concise but extremely thorough chapter on the ARPAnet and how it became the Internet.

(114) Williams 1979, p399.

(115) Williams 1979, p402.

We should note that the swapping of information between magnetic core and drum was called "automatic" swapping, meaning that the machine had virtual memory. Prior to this invention, virtual memory had meant that the programmer had had to look up the relevant information on the drum.

(116) Campbell-Kelly and Aspray 1996, p209; also see Mandl 1967.

(117) McCarthy 1959, 1983.

(118) Rodgers 1969.

(119) McCarthy 1959, 1983, Levy 1984 The system interrupt features of the Atlas were adopted in large commercial computer systems, specifically in the IBM System/360, in 1964; Williams 1979, p402. McCarthy's early note mentions that timesharing: "is being developed for various advanced computers, e.g., Stretch TX2, Metrovich 1010, Edsac 3. I would not be surprised if almost all of it is available with the transistorized 709." (1959, Memorandum to P.M. Morse Proposing Time Sharing").

(120) Newell, McCarthy CBI interviews; Levy 1984.

There are a number of accounts of this narrative. First, from the horse's mouth:

McCarthy, John. "Reminiscences on the History of Time Sharing". <http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html>. 1983. and:

McCarthy, John. "Memorandum to P.M. Morse proposing Time Sharing". <http://www-formal.stanford.edu/jmc/history/timesharing-memo/timesharing-memo.html>. January 1, 1959. Also quoted in A Century of Electrical Engineering and Computer Science at MIT. K. Wildes, MIT Press 1985, p243.

Another source is Judy O'Neill's doctoral dissertation (the University of Minnesota, 1992) on the birth of timesharing. Other sources concerning the initial invention of timesharing include the various CBI Oral History Interviews, notably those of McCarthy, Jack Dennis and Fernando Corbato. Transforming Computer Technology studies this topic exhaustively, as does Levy 1984 and Tracy Kidder's The Soul of a New Machine 1981.

(121) Levy 1984, p67-68.

(122) Licklider CBI OH.

(123) Campbell-Kelly and Aspray 1996, p209.

(124) Hafner and Markoff p265; Hafner and Lyon 1996.

(125) Campbell-Kelly and Aspray 1996, p215.

- (126) **1969**: Multics project for timesharing; Bell Labs pulls out of project, after which Multics is adopted by Honeywell as a secure operating system for military machines; Hafner and Markoff, p266.
- (127) Campbell-Kelly and Aspray 1996, p215.
- (128) Rodgers 1969, p279; Campbell-Kelly and Aspray 1996, p215.
- (129) " ...As early as 1970, the computer utility was being described by some industry insiders as an illusion and one of the computer myths of the 60s. By 1971 several firms were in trouble. UCC saw its share price dive from a peak of 186 to 17 in a few months. The demise of the time sharing industry was to set back the computer utility dream for the best part of 20 years. While simple time sharing computer systems survived, the broad vision of the computer utility was killed off by two unforeseen and unrelated circumstances; the software crisis and the fall in hardware prices." Campbell-Kelly and Aspray 1996, p218.
- (130) Levy 1984, p123-125.
- (131) Campbell-Kelly and Aspray 1996, p209.
- (132) Martin Greenberger, " The Computers of Tomorrow", The Atlantic Monthly. (May 1964).
- (133) The Carnegie Institute of Technology became Carnegie-Mellon University when Carnegie Tech merged with the Mellon Institute of Technology in 1967. Later, CMU simply dropped the hyphen.
- (134) " A..I.. Themes in the Second Decade". Information Processing 1968: Proceedings of the IFIP Conference '68. Amsterdam: North-Holland 1969: 1968: 1008-1024.
- (135) Norberg and O'Neill 1992, p302.
- (136) Norberg and O'Neill 1992, p312.
- (137) M. Minsky 1962; printed in Computers and Thought.
- (138) M. Minsky 1962; printed in Computers and Thought. Ibid.
- (139) This topic is voluminous: any treatment will give it short shrift. This is especially true in light of several excellent full-length books, from respective different perspectives, given to it.

Mind and Hand: The Birth of MIT, by MIT President (1959-1966) Julius A. Stratton and Loretta H. Mannix clearly retires the genre of full-length histories of MIT. (The MIT Press, 2005).

Karl Wildes and Nilo Lundgren's A Century of Electrical Engineering and Computer Science at MIT 1882-1982 (The MIT Press, 1985) is more focused on these

specific topics, with extensive discussions of Whirlwind, SAGE, and timesharing among other topics.

We also refer to:

Atsushi Akera, Calculating a natural world: Scientists, Engineers and Computers during the rise of U.S. Cold War Research, 2008).

Hapgood, Fred. Up the infinite corridor: MIT and the technical imagination. Reading, MA: Addison-Wesley Pub. Co. 1993.

David Kaiser, ed., Becoming MIT: Moments of Decision. 2010.

Levy, Steven. Hackers: Heroes of the Computer Revolution. Garden City, N.Y.: Anchor Press-Doubleday. 1984. Delta Trade Paperbacks (Bantam Doubleday). 1994.

Leslie, Stuart. The Cold War and American Science: The Military-Industrial-Academic Complex at MIT and Stanford. New York: Columbia University Press. 1993.

(140) Stratton, Julius A., and Loretta H. Mannix. Mind and Hand: The Birth of MIT. The MIT Press. 2005.

Kaiser, ed., Becoming MIT. Hapgood. See also "mitfacts" on MIT's own website.

(141) Deborah Douglas, "MIT and War," in Kaiser, ed., Becoming MIT; p95.

(142) Leslie 1993, p15.

(143) David Kaiser, "Elephant on the Charles," in Kaiser, ed., Becoming MIT, p105. Regarding the 'Rad Lab", p107.

(144) Leslie 1993, p20.

(145) Dickson 1971, p153.

(146) Leslie 1993, p15.

(147) Pfeiffer 1962, p1.

(148) Fernando J. Corbato, CBI Interview 162, 1989.

(149) Kahn, CBI OH 192, 1990.

(150) Minsky, CBI OH 179, 1989.

(151) McCarthy, CBI OH interview 156.

(152) Patrick Winston CBI OB Interview 196, 1990.

(153) Between them, Hapgood's study of MIT, Wilde and Lindgren's illustrated history, and Levy's Hackers: Heroes of the Computer Revolution tell this story fully.

(154) Nightwork, 2003, p4. Nightwork, recently reissued by the MIT Press, also refers to the "IHTFP, Gallery", a website specifically devoted to MIT Hacks, at hacks.mit.edu.

(155) Hapgood 1997 and Levy 1984 describe this history very well.

(156) Levy 1984.

(157) Klahr and Kotovsky 1992, p166.

(158) Levy 1984; Larry Roberts CBI OH interview

(159) Levy 1984, p23.

(160) Norberg and O'Neill 1996, p10

(161) Norberg and O'Neill 1995

(162) CBI OH 179, 1989.

(163) CBI OH 164, 1989.

(164) Hapgood 1997.

(165) Stanford historian Fred Turner's book regarding the influence of the Counterculture on computing discusses this.

(166) R. Stallman, "EMACS: the Extensible, Customizable Self-Documenting Display Editor." MIT AI Lab AI Memo 519A. 26 March 1981, p20.

Stallman is more a pure computer scientist than an AI figure. Levy writes about Stallman at some length in his 1984 work. He is so dedicated to his work that in the late 1970s, when he became despondent over the political infighting that was spoiling the development of LISP machines, he would tell acquaintances that his wife had died.

(167) Levy 1984, p50.

(168) Levy 1984, p98-100.

(169) Levy 1984, p98-100.

Three Scientists and their Gods: A Search for Meaning in an Age of Information, by Robert Wright (1988) contains a lengthy biographical essay on Fredkin.

(170) CBI oral history interview. McCorduck also discusses Fredkin.

(171) Fredkin told this anecdote at his keynote address at the AAI-1997 in Providence, Rhode Island.

(172) Levy 1984, p50.

(173) Levy 1984, p40-41.

(174) Minsky 1988, p306

(175) Minsky 1988, p306

(176) Minsky, M." The Graphical Typewriter: A Versatile Remote Console Idea." MIT Project MAC. Memo 65. January 24, 1964

(177) Minsky 1988, p306.

(178) Fano, R.M." The MAC System: A Progress Report." MIT Project MAC. MAC-TR-12. October 9, 1964.

(179) "A system in which a particular device is used for two or more concurrent operations. Thus the device operates momentarily to fulfill one purpose then another, returns to the first, and so on in succession until operations are completed", Webster's NewWorld Dictionary.

(180) Fano 1964

(181) Fano 1964.

(182) Fano 1964.

(183) The Project Mac typewriter terminals were closely adapted from the 'golf ball' IBM Selectric office typewriter, at the time the state of the art.

(184) Fano 1964.

(185) Levy 1984, p123-125

(186) Hafner and Markoff 1991, p266

(187) John White." Time-Sharing Lisp for the PDP-6." MIT Project MAC. AI Memo 157. March 1968.

(188) Larry Krakauer." Producing Memos, using TJ6, TECO and the type 37 teletype." MIT Project MAC. AI Memo 164. September 1968. Also, see Peter Samson." PDP-6 TECO." Artificial Intelligence Project. Project MAC. Memo 81. July 1965." The original TECO (tape editor and corrector) was a PDP-1 program written by Daniel L. Murphy. With the arrival of a PDP-6 at project MAC, the need was seen for a similar program on the new machine, and PDP-6 Teco was written by Steward Nelson, J. Holloway and Richard Greenblatt."

(189) Richard Stallman," EMACS: the Extensible, Customizable Self-Documenting Display Editor." MIT AI Lab AI Memo 519A. 26 March 1981.

(190) It is possible to find even further degrees of modesty than this in Stallman's early writing. In several of the papers of the MIT AI Lab series, the author actually uses the third person, awkward as it may be, to announce his recent work: as in, " A program has been written that..." It almost seems un-American. " The development of EMACS followed a path that most authorities would say is a direct route to disaster. It was the continuous deformation of TECO into something which is totally unlike TECO... And during the whole process, TECO and programs containing TECO were the only text editors we had on ITS... this is no accident. EMACS could not have been reached by a process of careful design, because such processes arrive only at goals which are visible at

the outset, and whose desirability is established on the bottom line at the outset. Neither I nor anyone else visualized an extensible editor until I had made one, nor appreciated its value until he had experienced it. EMACS exists because I felt free to make individually useful small improvements on a path whose end was not in sight. ...

Richard Stallman," EMACS: the Extensible, Customizable Self-Documenting Display Editor." MIT AI Lab AI Memo 519A. 26 March 1981, p20).

(191) Greenblatt, Richard D.; Donald E. Eastlake, III; and Stephen D. Crocker." The Greenblatt Chess Program". Fall Joint Computer Conference, 1967: p801.

(192) Greenblatt, Eastlake, and Crocker 1967, p804

(193) Minsky, Marvin and Seymour Papert." 1968-1969 Progress Report." MIT AI Memo 200, p32

(194) Marvin Minsky." Remarks on Visual Display and Console Systems." MIT Project MAC. June 1963. Reprinted as AI Memo 162 July 1968. p5.

(195) Mentioned in Greenblatt 1967.

(196) R.M. Fano," The MAC System: A Progress Report." MIT Project MAC. MAC-TR-12. October 9, 1964.

(197) Fano 1964, p3.

(198) Ivan Sutherland, "Sketchpad: a Man-Machine Graphical Communication System." Proceedings- Spring JCC 1963: 329-346. Sutherland started out at Lincoln Labs and BBN, and later decamped to the University of Utah. See his later article, " Computer Displays," Scientific American (June 1970): p56-70.

(199) M. Minsky," Remarks on Visual Display and Console Systems." MIT Project MAC. June 1963. Reprinted as AI Memo 162 July 1968; p6).

(200) M. Minsky," Remarks on Visual Display and Console Systems." MIT Project MAC. June 1963. Reprinted as AI Memo 162 July 1968; p6.

(201)M. Minsky," Remarks on Visual Display and Console Systems." MIT Project MAC. June 1963. Reprinted as AI Memo 162 July 1968, p8.

(202) Bitmap: bit mapped graphics: "computer graphics that are stored and held as collections of bits in memory locations corresponding to pixels on the screen. Bit-mapped graphics are typical of paint programs, which treat images as collections of dots rather than as shapes. Within a computer's memory, a bit mapped graphic is represented as an array (group) of bits that describe the characteristic of the individual pixels making up the image. Bit mapped graphics displayed in color require several to many bits per pixel, each describing some aspect of the color of a

single spot on the screen." Microsoft Press Computer Dictionary. Second Edition. 1996;

(203) Minsky, Marvin." Remarks on Visual Display and Console Systems." MIT Project MAC. June 1963. Reprinted as AI Memo 162, July 1968, p8.

(204) M. Minsky," Remarks on Visual Display and Console Systems1.1" MIT Project MAC. June 1963. Reprinted as AI Memo 162 July 1968.

(205) M. Minsky," Television Camera-Computer Adapter; PDP 6 Device 770." MIT Project MAC. AI Project Memo 75. Memo MAC 11-215. January 9, 1965, p1.

(206) Warren Teitelman," Argus." MIT AI Project Memo 53. May 28, 1965. See also re handwriting recognition and visual; ch 6 of Cote The Search for the Robots.

(207) Minsky, Marvin and Seymour Papert." 1968-1969 Progress Report." MIT AI Memo 200, p33.

(208) Ibid., p33.

(209) Raphael 1976, p261-262.

(210) Raphael 1976, p261-262.

(211) MH-1. A Computer operated mechanical hand. Dissertation Ph.D., EE. MIT 1961. Presented at the Western JCC, May 1962. Re robotics in general during this period, see also A. Cardoza and Suzee J. Vlk, Robotics. Blue Ridge Summit, PA: Tab Books. 1985; see p75+ re engineering automation re this topic: Booth, Digital Computers in Action. 1965. R Malone. Rattray Taylor article re robotics and haptics work at MIT, p175; Ch8 of Cote, re industrial robotics during this period.

(212) Bernstein 1983, p18.

(213) Description of thesis of David L. Waltz in Minsky and Papert, 1968-1969 Progress Report, MIT AI Memo 200, p37.

(214) Minsky and Papert, 1968-1969 Progress Report, MIT AI Memo 200, p35. Also see re robotics and mechanical hands: M. Minsky," Design of the Hand." MIT AI Project. Memo 86. August 13, 1965; and Stewart Nelson and Michael Levitt." Robot Utility Functions", MIT Project MAC AI Memo 172. February 1969.

(215) Minsky and Papert, 1968-1969 Progress Report, MIT AI Memo 200: p38-39.

(216) Minsky and Papert." 1968-1969 Progress Report." MIT AI Memo 200: p38.

(217) The photocell is also called an electric eye or photoelectric cell. Its technical base was the photoelectric effect; the emission of electronics from materials exposed to energetic radiation such as light. The velocity of the emitted electrons depends on light

frequency, and their quantity on light intensity... (HarperCollins Dictionary of Electronics).

(218) Adolfo Guzman, "Decomposition of a Visual Scene into Three-Dimensional bodies." MIT Project MAC. AI Memo 171. January 1969.

(219) Minsky and Papert, 1968-1969 Progress Report, MIT AI Memo 200, p40; also AI Memo 178.

(220) Seymour Papert, "The Summer vision project." MIT Project MAC. Vision Memo 100. July 1966

(221) Minsky and Papert. "1968-1969 Progress Report." MIT AI Memo 200.

(222) Minsky and Papert. "1968-1969 Progress Report." MIT AI Memo 200, p10).

(223) Sussman, Gerald Jay, and Adolfo Guzman. "A Quick Look at Some of our Programs." Summer Vision Project. Internal Memo #102. July 20 1966.

(224) Sussman, Gerald Jay, and Adolfo Guzman. "A Quick Look at Some of our Programs." Summer Vision Project. Internal Memo #102. July 20 1966, p4.

(225) A. Guzman, "Decomposition of a Visual Scene into Three-Dimensional bodies." MIT Project MAC. AI Memo 171. January 1969.

(226) M. Minsky, and Seymour Papert. "1968-1969 Progress Report." MIT AI Memo 200, p12.

(227) Minsky and Papert. "1968-1969 Progress Report." MIT AI Memo 200, p10.

(228) Minsky and Papert. "1968-1969 Progress Report." MIT AI Memo 200. p4

(229) "1968-1969 Progress Report." MIT AI Memo 200, p40.

(230) This paper, and Minsky's 'Frames' paper and Patrick Winston's 'Arch-Learner', were gathered into a volume edited by Patrick Winston, The Psychology of Computer Vision in 1975.

(231) Norberg and O'Neill 1992, p312.

(232) Edward A. Feigenbaum, "A..I.. Themes in the Second Decade". Information Processing 1968: Proceedings of the IFIP Conference '68. Amsterdam: North-Holland 1969: 1968: 1008-1024.

(233) Carnegie-Mellon University was not formed as such until 1967, following the Carnegie Institute of Technology's acquisition of the women's college immediately adjacent to the CIT campus. The women's college building, the foyer of which is decorated with murals praising housewifery, now houses the fine arts classrooms.

(234) Newell CBI OH interview

(235) Denicoff 1988; McCarthy OH 156. Other monies from the Psychology Department at CIT went to Newell and Simon, mentioned in Newell CBI interview. Simon duly recorded this construction of institutional infrastructure in his autobiography, but made clear that he dislikes the petty aspects of politics. He devoted an entire chapter to high-level university politics, with a worldly disclaimer entitled "Why I am not a University President".

(236) EAF OH#14.

(237) Edward Feigenbaum, "What hath Herbert Simon Wrought" in Klahr and Kotovsky, eds. Complex information processing: the impact of Herbert A. Simon. 1989.

(238) Newell and Simon, 1976, p40.

(239) Simon 1991, p272

(240) Both of the following books constitute 'the right place': Gardner, The Mind's New Science, 1985, and Leahey and Harris, Human Learning, 1989.

(241) CBI interview.

(242) MOML, p223-224.

(243) The Introspectionists were a progressive school of psychology, founded in Germany late in the Nineteenth century. They are notable for their earnest reliance on written records of individuals' talking through their thoughts.

(244) See the author's discussion of Wiener, Bigelow, and Rosenbluth and the 'Behavioristic' method, in the previous volume.

(245) Psychological Review, 1958, Simon 1991, p270.

(246) See Simon and Feigenbaum 1964; FN: EPAM lived on even after Simon and Feigenbaum were through with it- it is currently in at least its fifth version. Simon, H.A., with E.A. Feigenbaum." A Theory of the Serial Position Effect: Effects of similarity, familiarization and meaningfulness in verbal learning". In MOT (orig. 1964): 114-123.

(247) "When I came to Carnegie Mellon in 1969, there was no activity here in the perceptual motor end of the spectrum. That was one of the reasons both Simon and Newell, I think, desired that I join them here." (Raj Reddy CBI OH interview 231, 1991).

(248) The article is A. Newell's "On the analysis of human problem-solving protocols". In Johnson-Laird and Watson, eds. Thinking: Readings in Cognitive Science. Cambridge, England: Cambridge University Press. 1977: 46-60.

(249) A **demon** is an impersonal process that does not necessarily respond to the input of a particular user. These include, for instance, returning electronic mail dispatches sent to incorrect addresses, or responding to incoming mail by telling the sender that the recipient is on vacation. A **wizard**, which may also be

called in current parlance an **agent**, is personal. Wizard is the more current parlance, and the author is not sure that it predates the 1990s. The term 'demon' has been used continuously in computer since Selfridge coined it.

(250) Newell 1962, In Nii, "Blackboard Systems", Handbook IV, p18.

(251) Forgy and McDermott 1977.

(252) Nii, "Blackboard Systems", Handbook IV, p19.

(253) Simon 1966.

(254) Ernst and Newell 1969.

(255) Charniak and McDermott 1985, p295.

(256) Simon 1991, p225.

(257) IEEE Expert, June 1993, p5.

(258) Simon, H.A., with Allen Newell and J.C. Shaw. " The process of Creative Thinking" in MOT. Orig. 1962.: 144-174.

(259) " Simon 1991, p272.

Simon, H.A." Scientific Discovery and the Psychology of Problem Solving". Models of Discovery. Boston, MA: D. Reidel. 1977.

(Originally published 1966). Also see Simon, H.A."

Motivational and Emotional Control of Cognition". MOT. Orig. 1967: 29-38.

(260) See e.g., Johnson-Laird, Roger Schank, Robert Sternberg, Howard Gruber, Czikszentmihalyi, etc.

(261) (1561-1626). This Bacon is not to be confused with Roger Bacon, three centuries before, who endorsed greater attention to Aristotle.

(262) Thagard, Computational philosophy of science, 1988; Thagard, Conceptual Revolutions, 1992; and Holyoak and Thagard. Mental leaps: analogy in creative thought, 1995.

(263) Representative works include:

Simon, Herbert. Models of discovery: and other topics in the methods of science. Dordrecht, Holland; Boston: D. Reidel. 1977. Boston studies in the philosophy of science; v. 54; Synthese library; v. 114.

Langley, Pat." Learning to Search: From Weak Methods to Domain-Specific Heuristics". Cognitive Science 9. 1985: 217-260.

Langley, Pat and Herbert Simon." The Central Role of Learning in Cognition". In Anderson, ed. Cognitive Skills and Their Acquisition. Hillsdale, NJ: Lawrence Erlbaum Associates. 1981: 361-380.

Langley, Pat; Simon, Herbert A.; Bradshaw, Gary L.; and Zytkow, Jan M. Scientific Discovery: Computational Explorations of the Creative Processes. Cambridge, MA: The MIT Press. 1987.

Langley, Pat and Simon, Herbert A." Applications of Machine Learning and Rule Induction". 10.94; forthcoming C-CACM.

Klahr, David, Pat Langley, and Robert Neches, eds. Production system models of learning and development. Cambridge, MA: The MIT Press. 1987.

(264) Simon 1991, p225.

(265) Barr and Feigenbaum 1981, p190-191.

(266) The scholarship on production systems includes Newell and Simon's Human Problem Solving (1972); a clear précis in Volume I of the Handbook of Artificial Intelligence; Models of my Life, Unified Theories of Cognition, and more discursive coverage in Klahr, Langley and Neches, or Unified Theories of Cognition.

(267) Post 1941; see discussion in (The Handbook of AI vI, p18-20

(268) Simon 1991, p225-227.

(269) Simon 1991, p227.

(270) Floyd had been using production systems for compilers; Crevier 1993, p150.

(271) Bruce Buchanan on the Dendral project. CBI Interview OH 230. 1991.

(272) Buchanan CBI interview 1989.

(273) Lindsay, Buchanan, Feigenbaum, and Lederberg, 1980.

(274) interview with Ted Shortliffe, June 10, 1994.

(275) EAF 1992.

(276) McCorduck 1979. EPAM was still in use in research on memory at CMU as of the late 1990s. See Richman et al. 1995.

(277) Norberg CBI Interview with Buchanan, 1991

(278) Buchanan CBI Interview OH 230. 1991; Buchanan moved to the University of Pittsburgh in 1989

(279) Buchanan CBI OH 230, 1991.

(280) Buchanan CBI OH 230, 1991.

(281) Barr and Feigenbaum 1982 vII, p102

(282) Bowden, "Digital Computation and the Crystallographer", p203-209.

(283) Robert K. Lindsay, another major collaborator, had also done his doctoral work on inference and natural language understanding with Simon (Models of my Life p219, p227).

(284) Djerassi is more famous as the inventor of the birth control pill, in addition to his several redundant successes as a chemist, fiction writer, environmentalist, and art collector; see his autobiography.

(285) Personal interview, Robert Englemore of Stanford KSL, July 1994; also Applied Artificial Intelligence Research: The Heuristic Programming Project, 1990, p1).

(286) Bruce Buchanan, CBI Interview OH 230. 1991

- (287) Feigenbaum 1992 paper.
- (288) Buchanan, Bruce G. CBI Interview OH 230. 1991
- (289) Barr and Feigenbaum 1982, p106.
- (290) McCorduck 1979, p283.
- (291) Buchanan CBI interview.
- (292) Hayes-Roth et al., p8. Sources concerning the Dendral Project include: the pertinent CBI interviews; Buchanan, Sutherland, and Feigenbaum 1969; Buchanan and Feigenbaum 1978. Lindsay, Buchanan, Feigenbaum, and Lederberg, 1980; the Handbook, Volume II; the Stanford Knowledge Systems Laboratory's Applied Artificial Intelligence Research: The Heuristic Programming Project, 1990; and Hayes-Roth et. al.'s Building Expert Systems.
- (293) KSL, Applied Artificial Intelligence Research: The Heuristic Programming Project 1990, p46.
- (294) Handbook VII, p110.
- (295) EAF IJCAI 1977. I. Goldstein and S. Papert refer to the change in emphasis as "the shift to the knowledge-based paradigm"; "Artificial Intelligence, Language, and the Study of Knowledge" Cognitive Science (1977) 1: 1.
- (296) Lowood 1988.

(297) Scholarly literature on this history abounds. Much of it has come, not surprisingly, from Stanford and Berkeley. Regional Advantage: Culture and Competition in Silicon Valley and Route 128 (1994) by Annalee Saxenian, studies the regional innovative zones of Silicon Valley and Route 128 in detail. Stanford University's director of Special Collections Henry Lowood wrote an extensive monograph on Stanford's development as a tech innovator (1988).

Another work is Stuart Leslie's article, "How the West was Won: The Military and the Making of Silicon Valley" In William Aspray, ed., Technological Competitiveness: Contemporary and Historical Perspectives on the Electrical, Electronics and Computer Industries. New York: IEEE Press. 1993: 75-89. This and his book, The Cold War and American Science: The Military-Industrial-Academic Complex at MIT and Stanford (1993), consider the Cold War's role in shaping these institutions.

More recently this specific topic has been studied by Christophe LeCuyer in Making Silicon Valley, based on his dissertation on the history of Varian Corporation. And in Cold War Science and the Search for the Next Silicon Valley (2005) historian Maureen O'Mara considered whether and how the sociological character of the Valley could be replicated in other places.

All of the following are excellent historical analytical overviews:

Arthur Norberg, "The Origins of the Electronics Industry on the Pacific Coast", 1976. Proceedings of the IEEE. 64 (1976): 1314-1322. <http://www.leland.stanford.edu/class/history262/>

Henry Lowood, " From Steeples of Excellence to Silicon Valley: The Story of Varian Associates and Stanford Industrial Park". HYPERLINK [http://www-leland.stanford.edu/class/history262/;](http://www-leland.stanford.edu/class/history262/)

Tom Wolfe, "The Tinkerings of Robert Noyce: How the Sun Rose on the Silicon Valley," Esquire Magazine, December, 1983: 346-374.

HYPERLINK <http://www-leland.stanford.edu/class/history262/> <http://www-leland.stanford.edu/class/history262/>

U.C. Berkeley Geography professor Richard Walker discussed the deeper roots of the region's innovative abilities in fascinating detail in, "Another round of globalization in San Francisco". Urban Geography 1996: P60-94), heavily referenced in this chapter.

Copious archival materials substantiate the early history of all of these institutions. The Stanford University Archives and the oral history of the DARPA IPTO, available in interview transcripts from CBI, are wonderful sources. The history is also told implicitly, albeit never gathered as such, in the research proposals and memoranda of the individuals at the universities, and more explicitly in Norberg and O'Neill's 1997 book.

(298) Walker 1996.

(299) Walker; Norberg 1976, p1315.

(300) Morgan 1967, p14.

(301) Morgan 1967, p19.

(302) Morgan 1967, p29- the transmission from San Jose continues as KCBS).

(303) Morgan 1967, p19.

(304) Morgan 1967, p86.

(305) Morgan 1967, p56.

(306) Norberg 1976, p1316.

(307) The H-P garage was located at 376 Addison Street in Palo Alto (Morgan 1967, 121-122); Farnsworth Electronics was located at 202 Green Street in San Francisco's North Beach (ibid. p105).

*see NASA Historical Data Book. Volume IV. Nasa Resources 1969-1978. Washington, D.C.: Nasa History Office. 1994.

(308) LeCuyer 2006.

(309) Farnsworth Electronics was located at 202 Green Street in San Francisco's North Beach (Morgan 1967, p105).

(310) Lowood 1988.

(311) Clark Kerr, THE MAKING OF THE MULTIVERSITY.

(312) Morgan 1967, p187.

(313) Lowood 1988; also refer to recent biography of Shockley.

(314) Intel book reference.

(315) Leslie 1993, p45.

(316) Leslie 1993, p12.

(317) <http://www-soe.stanford.edu/soe/hist.html>.

(318) "computer science" (CBI Interview #15, 1979).

(319) Engelbart 1986 interview.

- (320) CBI interview with Alexandra Forsythe, 1979.
- (321) 1965. EAF CBI oral history project interview.
- (322) Feigenbaum CBI OH interview.
- (323) EAF CBI interview.
- (324) EAF Interview 1979, CBI OH #14.
- (325) See also: <http://www-db.stanford.edu/pub/voy/museum/Forsythenews.html> George forsythe his vision and its effects; David Salisbury, Stanford News Service.
- (326) Hilts 1988.
- (327) CBI interview.
- (328) Personal conversation March 1994.
- (329) McCorduck 1979.
- (330) EAF Interview 1979, CBI OH #14.
- (331) EAF Interview 1989, CBI #157.
- (332) Buchanan 1988, p313).
- (333) McCorduck 1979, p220
- (334) Hilts 1982, p284.
- (335) Hilts 1982, p259.
- (336) Hilts 1982, p261.
- (337) Hilts 1982, p261.
- (338) McCarthy archives (SC524); Box 1, folder 23; Human Rights 1975-1978, Committee of Concerned Scientists; also box 3, folder 27, Soviet Jews 1975 1978.**
- (339) Hilts 1982, p283.**
- (340) McCarthy archives (SC524); box 2, folder 13.
- (341) In 1980-1981; Box 1, Folder 18 of the McCarthy archives, SC524.
- (342) McCarthy archives, SC524, box 3, folder 19.
- (343) From 1970-1977; McCarthy archives, SC524, box 3, folder 6.
- (344) Personal communication, Avron Barr, 7.25.94.
- (345) CBI Oral History #156.
- (346) Generally speaking, pure nerds are concerned with the abstract administration of individual rights, and are often concerned with moral in an abstract sense. But they don't like power, and they exert it minimally even when they are qualified to do so. They are also often quite kind. Hans Moravec told Daniel Crevier that he hated to fire even incompetent people, and would hide in his office for months after having had to do so (Crevier 1993).
- (347) McCorduck 1979, p220.
- (348) Raj Reddy CBI OH interview 231, 1991.
- (349) Raj Reddy CBI OH interview 231, 1991.

(350) Reddy re who was at the Stanford AI Lab in the beginning: " McCarthy and Feigenbaum were co-PIs [at the AI Lab at Stanford]]. Les Ernest was the manager...Let's see, other major investigators... Raj Reddy had just finished, and was an assistant professor working on speech understanding. He had a group there that included Lee Erman. Jerry Feldman had just finished, and was there. Kent Colby was there working on PARRY... There was a robotics group. Dick Scheinman was there. And there was the Les Ernest project too. There was work on chess. Barbara Liscov Hubermann was an early McCarthy student. There was quite a bit of systems work going on to make the PDP-6 into a research machine that was reliable. Jerry Feldman came soon... I can't remember the timing very much. And then some time around 1970 or 1971 Terry Winograd came...It was an exciting place." (Buchanan, Bruce G. CBI Interview OH 230. 1991).

(351) Buchanan 1988, p314.

(352) Buchanan 1988, p314.

(353) Nils Nilsson, CBI OH Interview 155, 1989.

(354) Modal logic had been used in philosophy earlier in the century; Hayes and McCarthy 1969, p492.

(355) Nils Nilsson, CBI OH Interview 155, 1989.

(356) Weyhrauch, Richard W." Prolegomena to a theory of mechanized formal reasoning," Artificial Intelligence 12 (1,2, 1980): 133-170. Reprinted in Readings in knowledge representation; by Brachman and Levesque.

(357) Buchanan 1988, p317.

(358) This practice is not unheard of at graduate schools. For instance, in the summer of 1994, Stanford University Provost Dr. Condoleeza Rice issued a memorandum to research lab managers, suggesting that the lab managers notify campus security of their own 'resident' graduate students.

(359) Buchanan 1988, p315.

(360) Buchanan 1988, p314; documents are in the archive papers.

(361) Ray Reddy CBI OH interview.

(362) Buchanan 1988, p314,

(363) JMC Archives box 2, folder 6,7,8,9; LOTS Low Overhead Time Sharing System, 1979.

(364) JMC Archives [p2-5], Work of the Stanford AIL in 1973: 12 January 1974, box 2, folder 6,7,8,9.

(365) [p2-5], Work of the Stanford AIL in 1973: 12 January 1974; box 2, folder 6,7,8,9.

(366) [2-5] Work of the Stanford AIL in 1973: 12 January 1974; box 2, folder 6,7,8,9.

(367) [[2-5] Work of the Stanford AIL in 1973: 12 January 1974; box 2, folder 6,7,8,9.

(368) Discussion of the Arpanet Protocols Project, p2-6] Work of the Stanford AIL in 1973: 12 January 1974; box 2, folder 6,7,8,9.

(369) Buchanan 1988, p317.

(370) Buchanan 1988, p317.

(371) Buchanan 1988, p317.

(372) Levy 1984, p141-145.

(373) McCarthy archives SC524, box 1, folder 17.

(374) Box 2, folder 24; McCarthy archives SC524.

(375) Leslie 1993, p243. Also see SRI website.

More generally, materials concerning SRI include the following sources: Paul Dickson's Think Tanks (1971); Engelbart, Douglas. Interviews 1 through 4, (1987-1988); Leslie, Stuart. The Cold War and American Science: The Military-Industrial-Academic Complex at MIT and Stanford (1993); Nilsson, Nils (1988); and Rheingold, Howard. Tools for Thought: The People and Ideas behind the Next Computer Revolution (1985).

(376) Dickson 1971, p204; Engelbart 1987.

(377) Engelbart 1987, Interview 1.

(378) Dickson 1971, p204.

(379) Dickson 1971, p208.

(380) Douglas Engelbart's own interviews are at Stanford's sul site, under hasrg / histsci / ssvoral / engelbart. Interview 1, December 19, 1987.

(381) McKenney et al. 1995, p50-70; see Engelbart, Grosch, SRI website.

(382) McKenney 1995, p47.

General Electric later bought from BOFA the rights to the MICR technology, and then developed the dedicated computers for processing data embedded with this type. The SRI 'techies' who were hired to design the computers refused to leave Palo Alto for Arizona, so the GE project proved to be another computing first in the Bay Area. Grosch 1992, p247.

(383) Nilsson 1985 in Engelmores ed., p318-320.

(384) Nilsson 1985 in Engelmores ed., p320.

(385) 1963; Nilsson 1988 CBI OH.

(386) Nilsson 1985 in Engelmores ed., p319.

(387) Nilsson 1985 in Engelmores ed., p319-320

(388) Documents re Shakey the Robot:

Nilsson, Nils." A Hierarchical Robot Planning and Execution System." SRI. April 1973. Annual Report and Technical Note on SRI project 1187. NTIS report. AD 761 641.

Fikes, Richard A., et al." Learning and Executing Generalized Robot Plans." September, 1972. SRI. Army Research Office- Durham AD 765 242.

Hart, Peter E.; Nils Nilsson; and Ann E. Robinson." A Causality Representation for Enriched Robot Task Domains." SRI. ONR Report on SRI Project 1187. December 1971.

Klahr, Philip, and Donald Waterman, "Artificial Intelligence: A RAND Perspective", The AI Magazine, Summer 1986, 56-64. ? p22 ???
- they also made a movie of this- cite Nilsson re Shakey.

(389) Nilsson 1985 in Engelmores ed., p319-320

(390) Nilsson NTIS report 1973. AD 761 641.

Also re QA4: Rulifson, Derksen, and Waldinger, "QA4, A Procedural Calculus for Intuitive Reasoning," SRI AI Center Technical Notes 73, November 1973.

(391) Nilsson 1973 NTIS report AD 761 641. p7

(392) Handbook, vVIII, p523.

(393) Darrach, Brad." Meet Shakey, the first electronic person," Life magazine (November 20, 1970): 58.

(394) Nils Nilsson, CBI OH Interview 155, 1989.

(395) Nils Nilsson, CBI OH Interview 155, 1989

(396) Stanford Archives Engelbart oral history 1, 1987.

(397) Stanford Archives Engelbart oral history Interview #2.

(398) Writing in 1971, the science journalist Dickson expresses amazement at the dexterity which the seventy-seven ARC researchers had acquired in manipulating the mouse. This seems quaint, since today countless millions of people have acquired such dexterity.

(399) Stanford Oral History Interview 3.

(400) Stanford Archives Engelbart oral history 1, 1987. Interview 3.

(401) Engelbart Stanford OH Interview.

(402) SRI presented Engelbart with \$10,000 for the computer mouse. After he accepted the check, he discovered that this permanently precluded his ever receiving more money for what must be an unfathomably lucrative artifact. (Oral History interview 3).

Engelbart later worked at Tymshare, then at the Bootstrap Institute. See Douglas Engelbart, <http://www-sul.stanford.edu/depts/hasrg/histsci/ssvoral/engelbart/engfmst2-ntb.html> Interview 1, December 19, 1987. Also Rheingold 1984.

(403) The Reserve Officers' Training Corps.

(404) For all of the students' objections to the "military-industrial complex" and its undue role in American universities, the term itself was coined by President Dwight Eisenhower.

- (405) Nelkin 1972, p7.
- (406) Leslie 1993, p243.
- (407) Leslie 1993, p243.
- (408) Dickson 1971, p208.
- (409) Stanford Report vXXX, no.12, January 7, 1988.
- (410) Hilts 1982, p261.
- (411) Hilts 1982, p261.
- (412) Stuart Leslie." 'Time of Troubles' for the Special Laboratories", In Kaiser, David, ed. **Becoming MIT: Moments of Decision**. MIT Press, 2010: p124.
- (413) See Leslie 1993, p235-241.
Dorothy Nelkin's The University and Military Research: Moral Politics at MIT (1972) studies this as well:
"In May, 1970 Howard W. Johnson, president of MIT, announced to his faculty that the MIT Corporation could no longer continue to manage the Instrumentation Laboratory... The name of the lab was then changed to the Charles Stark Draper Lab... [In] 1969 the Instrumentation Lab had received \$54.6 million from DOD and NASA"
Nelkin 1972, p3-20.
- (414) Levy 1984, p133-135.
- (415) Kent State University in Ohio had been shaken by increasingly violent student demonstrations over the course of 1970. The University administration responded by calling the Ohio National Guard in to restore order. On May 4, 1970, the Guard shot and killed four undergraduates during the aftermath of a particularly contentious protest. Dramatized by one of the most famous photographs of the decade, showing a young woman mourning a dead companion's body, this event set off further unrest all over the United States.
- (416) MOML, p282.
- (417) MOML p282-289.
- (418) *ibid.*, p283.
- (419) Testimony of ARPA Director Dr. R.L. Sproull, DoD hearings, March 1964, p138.
- (420) Testimony of ARPA director Dr R.L. Sproull, DOD hearings before Congress April 1965, p536.
- (421) Bennett, Edward; James Degan, and Joseph Spiegel, eds. Military Information systems: the design of computer aided systems for command. New York: Praeger. 1964: p127.
- (422) DOD 1965 hearings.
- (423) U.S. Senate Subcommittee hearings, 1969.
- (424) The text of the law is: " None of the funds authorized by this Act may be used to carry out any research project or study

unless such project or study has a direct or apparent relationship to a specific military function or operations."

That provision became law and the same provision now appears as Section 204 of the military authorization bill reported the Senate for FY1971, but does not appear in the bill reported in the House." Penick et al 1972, p343.

(425) During the same year, Mansfield proposed "rechanneling the public resources for basic science through the civilian agencies: a new goal for science policy". National Science Policy, H con res 666. Hearings before the Subcommittee on Science Research and Development of the Committee on Science and Astronautics U.S. House of Representatives, 91st Congress 2 session 1970, p604-609; (Penick et al 1972, p 338-340).

(426) Penick et al 1972, p346.

(427) FY1975 HR hearings on Defense Appropriations, p649.

(428) Penick et al 1972, p346.

(429) JMC, CBI interview 156.

(430) Minsky CBI OH interviews. Apparently, neither had been exceptionally eager to carry out administrative chores in the first place. According to Minsky, "We worked so well together that for a decade, we each could run the laboratory effortlessly, leaving the other to decide what must be done." Minsky 1988, p306.

(431) Patrick Winston, CBI OH #196, 1990.

(432) Dertouzos, OH 164, 1989.

(433) Norberg and O'Neill 1996, p37.

(434) Heilmeier CBI Interview OH22, 1991; also Congressional hearings).

(435) Norberg and O'Neill, p35-40.

(436) Heilmeier CBI OH226, 1991.

(437) Heilmeier CBI OH226, 1991.

(438) Heilmeier CBI OH226, 1991.

(439) Saul Amarel CBI interview #176, 1989. The JASONS are a group of leading U.S. physical scientists to who devote their attention to problems of science and national security. The Jasons (named after Jason of Greek mythology) were organized originally in 1960 at the IDA w the support of the then Director of DR&E, Dr. Herbert F York. See, H F York, Making Weapons, Talking Peace, Bason Books, NY 1987. (Van Atta et. al 1990, p21-22.

(440) Dickson 1971, p141-143.

(441) DDR&E also asked an external review group to assess the DARPA AI programs in the 1970s; their conclusions were parallel to Heilmeier's. Communication from Dr A Flax, IDA, 2/ 90." (Van Atta et. al 1990, p21-11); Also Heilmeier CBI OH226, 1991.

(442) Minsky CBI OH 179, 1989.

- (443) CBI OH Interview 155, 1989; also J. McCarthy, CBI OH 156.
- (444) 1988, p322
- (445) Engelbart OH, 1987.
- (446) Newell, Reddy CBI OH.
- (447) personal conversation, Carl Hewitt, Palo Alto, July 4 2012.
- (448) FY1978 hearings, p63.
- (449) FY 1970 hearings, p134.
- (450) Statement of Heilmeier to Congress, 1977.
- (451) George H. Heilmeier Interview, 27 March 1991, OH226.
- (452) Buchanan OH CBI interview 1989.
- (453) Speaking historically, one could say that HASP and HARPY were the second versions of blackboards. Oliver Selfridge's Pandemonium, a decade or so earlier, was in fact a blackboard of shrieking demons, albeit lacking heuristics or production rules.
- (454) Congressional Hearings on Department of Defense Appropriations for 1966. March 1965, p1-2
- (455) Senate Subcommittee Hearings on Department of Defense Appropriations for FY1980. Part 4, Procurement/ RDT&E. S181-40. February 21, 1979. DDR&E. Testimony of DARPA Director Robert Fossum.
- (456) Testimony of ARPA Director Dr. R.L. Sproull, DoD hearings, March 1964, p156.
- (457) Testimony of ARPA Director Dr. R.L. Sproull, DoD hearings, March 1964, p156); Dept of Defense Appropriations for FY**1965**, Hearings before a Subcommittee of the Committee on Appropriations, House of Representatives. **Part 5: RDTE**, Appropriation language, testimony of Members of Congress, organizations and interested individuals, (88) H2062-0B, 88th congress, 2d session, 6, 11-13, 16-19 March 1964, p127.
- (458) 1967 documents, p138.
- (459) **1971 budget**; p726.
- (460) FY1985 House Committee hearings on DoD appropriations p490.
- (461) Testimony of Dr R.L. Sproull, ARPA director, DOD hearings before Congress April 1965, p523.
- (462) Congressional Hearings for FY1968, p138.
- (463) Statement by Dr. Eberhard Rechtin, ARPA director, 1971, p425.
- (464) p134.
- (465) 1972 documents, testimony of Stephen Lukasik, p745.

(466) Hearings before a Subcommittee of the Committee on Appropriations, United States Senate. 95th Congress, First session on HR 7933. Department of Defense Appropriations for FY1978. February 4, 1977. 77 S181-27.1. Part 5, RDT&E, p74.

(467) Hearings before a Subcommittee of the Committee on Appropriations, United States Senate. 95th Congress, First session on HR 7933. Department of Defense Appropriations for FY1978. February 4, 1977. 77 S181-27.1 Part 5, RDT&E. Statement submitted by Dr George H. Heilmeier. 44-172.

(468) Buchanan, Bruce G. CBI Interview OH 230. 1991.

(469) Vinton G. Cerf, CBI OH 191, 1990.

(470) Michael Dertouzos, CBI OH 164, 1989.

(471) CBI OH.

(472) Hapgood 1993, p129.

(473) Hapgood, Infinite Corridor book; Stewart Brandt.

(474) This chapter does not address certain other insider-outsider critiques, specifically that of Terry Winograd and Fernando Flores (1987+). Nor do we include the work of philosopher John Searle, or the deep thoughts of 'neurophilosophers' Patricia and Paul Churchland and their colleagues. Neurophilosophy generally speaking appeared on the scene later than the ambit of this book.

(475) McCorduck 1979, p173.

(476) How could someone with this attitude survive at MIT for several years, one wonders ?

(477) Pierce, John R. "What computers should be doing" [1962]. In Pierce, Shannon, Rosenblith, and Bush, "What Computers Should be Doing" in Greenberger, ed. 1962: 290-325.

(478) Dreyfus, H. and S., remarks in General Discussion, Greenberg ed. 1962, p321-322.

The lecture text is found in: M. Greenberger, ed. Management and the Computer of the Future. New York: MIT Press and Wiley. 1962; the same text was reissued in 1965 as Computers and the World of the Future.

(479) rationality of physics" (McCorduck 1979, p205.

(480) Baumgartner and Payr eds. 1995, p71.

(481) Models of My Life, p274.

(482) McCorduck 1979, p192.

(483) Mind Over Machine (1985).

(484) Herbert Simon discussed this matter in *Models of my Life*, as does McCorduck.

This means of acquiring a job infuriates Herbert Simon: " He'd had no connections with RAND before or since; he had no technical background for this at all. But the fact that he was a consultant

at RAND immediately gave him credibility. I was about to say I don't mind being criticized; of course I mind being criticized. But you know that's fair game and I can play it the way the politicians play it. But what I resent about this was the RAND name attached to that garbage. That was really false pretenses." (McCorduck 1979, p192).

(485) publication for nearly the entire academic year (MOML). But "Alchemy and AI" was out of the bag by mid-1965 (Rand Corporation 1965, P-3244).

(486) WCCD, ' 1972; Revised Edition 1979; Second Edition with new introduction, 1992) and other works (Mind over Machine, 1985).

(487) "Alchemy and AI" was subsequently published in Reading for Liberal Education, Holt Rhinehart and Winston 1970. Also see "Artificial Intelligence," Annals of the American Academy of Political and Social Science. 1975.

(488) Is the attitude evidenced in WCCD specific to the Dreyfus brothers or more generic to the profession ? It seems that philosophers often have the spite of gossip columnists and the memories of elephants: Karl Polanyi even specifically cited the words of Bukharin in the 1930s as the spark that set off his lifelong polemic against positivism; The Tacit Dimension, 1967.

(489) "Phenomenology: a term that emerged in the 18th century in the writings of Johann Heinrich Lambert (1728-1777) and Kant to denote the description of consciousness and experience in abstraction from consideration of its intentional content (see intentionality). In Hegel, phenomenology is instead the historical enquiry into the evolution of self-consciousness developing from elementary sense experience to fully rational free thought processes capable of yielding knowledge. The term in the 20c is associated with the work and school of Husserl. Following Brentano, Husserl realized that intentionality was the distinctive mark of consciousness, and saw in it a concept capable of overcoming traditional mind body dualism. The study of consciousness, therefore, maintains two sides; a conscious experience can be regarded as an element in a stream of consciousness, but also as a representative of one aspect or profile of an object. In spite of Husserl's rejection of dualism, his belief that there is a subject matter remaining after **epoche* or bracketing of the content of experience, associates him w the priority accorded to elementary experiences in the parallel doctrine of phenomenalism, and phenomenology has partly suffered from the eclipse of that approach to problems of experience and reality. However, later

phenomenologists such as Merleau-Ponty do full justice to the world involving nature of experience." (Blackburn 1994).

The author admits that she does not quite 'get' phenomenology, or perhaps she gets it and is astounded at how lightweight it is. Apparently she is not alone. While bidding goodbye to a graduate school classmate who had just 'Phinished', she noted a book on the topic on his desk and asked if he had ever really understood it. No, he replied with a smile. He then handed her a business card imprinted with his telephone number in India. "But if you ever figure it out, please call me collect and explain it to me !"

(490) Martin Heidegger (1889-1976): "German existentialist and social critic. Heidegger is probably the most divisive philosopher of the 20th century, being an acknowledged leader and central figure to many ('continental) philosophers and either a convenient example of meaningless metaphysics or else an apologist for Nazism, to other (analytical) thinkers.

" Freedom, existence in the world, inauthenticity, dread, guilt, and destiny therefore become the major themes. However, before they became the staple topics of existentialism they had a more sinister political embodiment; in 1933 Heidegger became Rektor of Freiburg, and his notorious Rektorsrede or inaugural speech "the role of the university in the new reich", was a call for Germany to move itself into the primordial realm of the powers of being, with the Nazi part in the vanguard."

"... when he writes that 'from a metaphysical point of view, Russia and America are the same, the same dreary technological frenzy, the same unrestricted organization of the average man' (An introduction to metaphysics, 1953), it is easy to forget that his contempt for the mass culture of the industrial age springs from a nationalistic and conservative elitism, rather than from any left wing or egalitarian illusions." Blackburn, S. Oxford Dictionary of Philosophy.

(491) Fuchs, p54.

(492) McCorduck 1979, p197.

(493) 1979, p235.

(494) ` The body contributes three functions not present and not as yet conceived in digital computer programs;

1 the inner horizon that is the partially indeterminate predelineated anticipation of partially indeterminate data (this does not mean the anticipation of some completely determinate alternatives or the anticipation of completely unspecified alternatives which would be the only possibly digital implementation);

2 the global character of this anticipation which determines the meaning of the details it assimilates and is determined by them

3 the transferability of this anticipation from one sense modality and one organ of action to another. All these are included in the

general human ability to acquire bodily skills. Thanks to this fundamental ability an embodied agent can dwell in the world in such a way as to avoid the infinite task of formalizing everything;"' [Dreyfus 1972, p167].

(495) Intentionality: The directedness or aboutness of many if not all conscious states. The term was used by the scholastics but revived in the 19th century by Brentano. Our beliefs thoughts wishes dreams and desires are about things. Equally the words we use to express these beliefs and other mental states are about things. The problem of intentionality is that of understanding the relation obtaining between a mental state or its expression and things it is about. A number of peculiarities attend this relation. First if I am in some relation to a chair for instance by sitting on it then both it and I must exist. But while mostly one thinks about things that exist sometimes (although this way of putting it has its problems) one has beliefs hopes and fears about things that do not as when the child expects Santa Claus and the adult fears Zeus. Secondly if I sit on the chair and the chair is the oldest antique in London then I sit on the oldest antique in London. But if I plan to avoid the mad axe man and the mad axe man is in fact my friendly postman, I do not therefore plan to avoid my friendly postman...

Intentional relations seem to depend on how the object is specified, or as Frege puts it the mode of presentation of the object. This makes them quite unlike the relations whose logic we can understand by means of the predicate calculus and this peculiarity has led some philosophers notable Quine to declare them unfit for use in serious science. More widespread is the view that since the concept is indispensable, we must either declare serious science unable to deal w the central feature of the mind or explain how serious science may include intentionality. One approach is to suggest that while the linguistic forms in which we communicate fears and beliefs have a two faced aspect, involving both the objects referred to and the mode of presentation under which they are thought of, we can see the mind as essentially directed onto existent things, and extensionally related to them. Intentionality then becomes a feature of language rather than a metaphysical or ontological peculiarity of the mental world. (Blackburn 1994).

(496) 1979, p272.

(497) HD 1979, p100.

(498) 1979, p272.

(499) " In the period between the invention of the telephone relay and its apotheosis in the digital computer, the brain, always understood in terms of the latest technological inventions, was understood as a large telephone switchboard or, more recently, as an electronic computer. This model of the brain was correlated w work in neurophysiology which found that neurons fired a somewhat

all or nothing burst of electricity. This burst or spike was taken to be the unit of information in the brain corresponding to the bit of information in a computer. This model is still uncritically accepted by practically everyone not directly involved with work in neurophysiology, and underlies the naive assumption that man is a walking example of a successful digital computer program." Dreyfus 1979, p159.

(500) " The theory we shall criticize claims that there is such a level- the information processing level- and that on this level the mind uses computer processes such as comparing classifying searching lists and so forth to produce intelligent behavior. This mental level, unlike the physical level, has to be introduced as a possible level of discourse. The issues involved in this discussion will therefore be philosophical rather than empirical. We shall see that the assumption of an information processing level is by no means so self-evident as the cognitive simulators seem to think- that there are good reasons to doubt that there is any information processing going on, and therefore reason to doubt the validity of the claim that the mind functions like a digital computer."

(Dreyfus 1979, p163).

(501) e.g., Anderson and Bowers 1973.

(502) Klahr, production systems article, 1973.

(503) " That he never said a computer couldn't do *that* were nearly Dreyfus' very words at the end of a chess match between himself and the MacHack program at MIT. His original statement in 'Alchemy and AI' had been ambiguous: no chess program can play even amateur chess. Did he mean *ever* ? That is certainly the sense understood by *The New Yorker* in its June 11 1966 'Talk of the Town' which reported on Dreyfus paper with even more than its usual smugness. Subsequently mated by MacHack, Dreyfus wrote in his book, "Embarrassed by my expose of the disparity between their enthusiasm and their results, AI workers finally produced a reasonably competent program. R Greenblatt's program called MacHack did in fact beat the author, a rank amateur." His footnote to this statement goes on to explain that he did not mean a computer could never play even amateur chess that he was giving a correct statement of the art at the time he wrote, 1965. On the other hand, Richard Greenblatt might dispute the assertion that he conceived MacHack in a flush of mortification over any expose." (McCorduck 1979, p199).

(504) Dreyfus 1979, p26.

(505) Crevier 1993, p123.

(506) Papert also wrote a paper on the topic. Papert, S." The Artificial Intelligence of Hubert Dreyfus: A Budget of Fallacies". AI Memo 154, MIT AI Lab 1968.

(507) McCorduck 1979, p197

(508) Greenberger 1962 ed.

(509) Born in Berlin, Weizenbaum moved to the United States in 1936. He served in the Army during the Second World War, and studied mathematics at Wayne University, now Wayne State University, in Detroit. He was a computing consultant at SRI during the late 1950s, and from 1963 through 1988, Weizenbaum taught at MIT as professor of Computer Science. Weizenbaum, Joseph. "The Myth of the Last Metaphor". See Baumgartner and Payr eds. Speaking Minds. 1995: 249-264.

(510) Baumgartner and Payr, eds.

(511) Communications of the ACM, C-ACM 1966.

(512) Weizenbaum, Joseph. "ELIZA: A computer program for the study of natural language communication between man and machine", C-ACM 1 (1966): 36-45.

(513) Weizenbaum 1976, p5-6.

(514) Weizenbaum 1976, p5.

Sagan, Carl. In "In praise of robots," Natural History LXXXIV, 1 (January 1975): p10-18.

(515) Sagan 1975, p10.

(516) Levy, p138.

(517) Weizenbaum 1995, p260.

(518) Simon 1991, p275.

(519) **The murderous HAL computer in Kubrick's 2001: A Space Odyssey.**

(520) Darrach 1970, p66.

(521) Darrach 1970, p68.

(522) Stork 1997, p19.

(523) Life Magazine December 10, 1970, p9.

(524) Hapgood 1974, p38); Fred Hapgood, "Computers Aren't so Smart After All" The Atlantic 234 (August 1974): 37-45.

(525) Sagan 1975, p10. **Sagan, Carl.** "In praise of robots," Natural History LXXXIV, 1 (January 1975): p10-18.

(526) This extended beyond the very public recrimination of the nuclear cold warriors, which had been intermittently ongoing as early as the late 1940s (e.g., most famously, by Norbert Wiener and J. Robert Oppenheimer). give some cites here- this is great stuff; Opp- the open mind and recent biography;

(527) M. Minsky and Seymour Papert's "Proposal to ARPA for Research on Artificial Intelligence at MIT, 1971-1972". p7-8.

(528) The Laboratory also engaged in work in theorem proving at the same time as it was a topic done at other

places. Minsky characterized this field as relatively sterile, since it purported to prove by logic without semantics: "... we feel that once such systems attempt to solve "real" problems, all such devices will fail as mere stop gaps; they do not help one to come to grips with the construction of the kinds of cognitive models we think are needed to solve hard problems by using accumulated knowledge and experience. M. Minsky and Seymour Papert." 1968-1969 Progress Report." MIT AI Memo 200. p22.

(529) The Artificial Intelligence Project- RLE and MIT Computation Center Memo 18. The text is not dated.

(530) John McCarthy Archives- Stanford Special Collections. History and State of the Art" is the best title I can find; JAF ?? Jerome A Feldman.

(531) Author's interview at Xerox PARC, July 1994.

(532) See Minsky ed., and the Handbook, vI, p284, p260. Bobrow, D.G." Natural language input for a computer problem-solving system", Ph.D Dissertation, Massachusetts Institute of Technology. 1964.

(533) Handbook, vI, p284, p260.

(534) Grammatical formalisms are described in terms of generative capacity, that is the breadth of the set of languages which they can represent. For instance, the most limited sort of grammar is a finite state grammar, in which the rewrite rules for parsing given sentences can cope with only a limited range of utterances. At the upper end of easing the restrictions on understanding of given utterances or sentences, recursively enumerable grammars are grammatical formalism which use unrestricted rules. Both sides of rewrite rules may have any number of terminal and non-terminal symbols. These are equivalent to Turing machines in their expressive power . Russell and Norvig 1995, p656.

(535) See Minsky, ed., and the Handbook.

(536) In a later publication with cognitive scientist X. Collins, Quillian included further relations; Handbook vIII, p38.

(537) References: Quillian, Ross." Semantic Memory". In Minsky, ed. Semantic Information Processing. Cambridge, MA: The MIT Press. 1968: 216-270. Quillian, Ross." Word Concepts: A Theory and Simulation of some basic semantic capabilities", Behavioral Science 12, 1967, 410-430.** Readings in knowledge representation; Brachman and Levesque.

(538) Bert Raphael." Operation of a Semantic Question-Answering System." Artificial Intelligence Project- RLE and MIT Computation Center. Memo 39. November, 1963.

- (539) Rochester, N." Symbol Manipulating Language." Artificial Intelligence Project- RLE and MIT Computation Center. AI Memo 5. 1958 Nov 18
- (540) James Slagle, 1961, at the AI Project.
- (541) Van Atta et. al 1990, p21-3; cites Discussion with Edward Lafferty 5.89
- (542) E. Fredkin, " Project MAC Progress Report IX, July 1971 to July 1972." MIT. AD 756 689. NTIS paper.
- (543) J. Moses." Symbolic Integration", MIT Project MAC. AI Project Memo 97. June 1966, p7.
- (544) Minsky and Papert, " 1968-1969 Progress Report." MIT AI Memo 200, p27.
- (545) Minsky and Papert 1968-1969 Progress Report, p26.
- (546) Winston CBI OH 196, 1990.
- (547) ARPA-IPTO Principal Investigator Meeting, Feb, 1971. TOC, list of Attendees, and misc short papers (BBN, SRI, MIT, MIT, CMU, SRI, RAND, SDC).
- (548) E. Fredkin, " Project MAC Progress Report IX, July 1971 to July 1972." MIT. AD 756 689.
- (549) See Stewart Brand's thorough The Media Lab: inventing the future at MIT, 1988.

(550) AILab: The AI Time-Sharing System.

" The experimental work of the AI group requires a high level of computer service for a limited number of users. Our system is based on time-sharing a two-processor machine PDP 6 and PDP 10 with a core memory of 2 to the 18 words of 36 bits. Normally, the programs of the active users remain in fast memory. This limits the number of users now, but a paging device to be completed early in 1970 should allow some expansion required by the maturation of certain projects, notably MATHLAB."

" The special requirements of the project led to a time sharing system with enough novel features to merit discussion [ref to document on this]; Besides the usual kinds of I/O and system calls, there are special calls to reduce overhead and to facilitate real time control. These include programs for operating the mechanical hands and computer eyes and other special remote control devices.

..

Because all user programs ordinarily reside in core, it is possible to switch between programs with great rapidity. Quanta of user time are short enough to allow program response to single typed characters without noticeable delays. .."

" A user may have many jobs running "simultaneously". Each user commands a tree of procedures; each can create and control

subordinate procedures; all have equal access to external devices and files."

... The secondary storage uses IBM 2311 discs and DEC microtape drives which are file structured in the same way. Users can establish symbolic links between file directories so that files can be shared by many programs and many users; these links can be chained.

The system has several operating stations. These include four text display devices, several teletype and external phone lines, a main console w DEC 340 display and several slave monitors around the laboratory, and a special console with controls for operating the eye hand system...

" A multiplexed digital analog system operates either in word or block mode, on call or automatically, as requested. Any number of users may simultaneously have analog access on different channels in different modes." Marvin Minsky and Seymour Papert." 1968-1969 Progress Report." MIT AI Memo 200.

(551) Minsky and Papert," Proposal to ARPA for Research on Artificial Intelligence 1970-1971." MIT AI Memo 185. December 1970.

(552) M. Minsky and Seymour Papert." Draft of a Proposal to ARPA for Research on Artificial Intelligence at MIT, 1970-1971", p39, quoted in What Computers Can't Do. Revised edition, 1979, p9.

(553) Terry A. Winograd (1946-) received his undergraduate degree at Colorado College, and then studied at University College, London. His Ph.D. in Applied Mathematics at MIT was conferred in 1970. He was Assistant Professor in Electrical Engineering at MIT, from 1971 through 1974, and then moved to Stanford's Department of Computer Science, where he is now Professor. (Winograd, "Computers and Social Values" In Baumgartner and Payr eds. Speaking Minds. 1995: 283-300.

(554) Crevier 1993.

(555) McCorduck 1979, p259.

(556) " The basic viewpoint guiding its implementation was that meanings (of words, phrases and sentences) can be embodied in procedural structures and that language is a way of activating appropriate procedures within the hearer. Thus, instead of representing knowledge about syntax and meaning as rules in a grammar or as patterns to be matched against the IP, Winograd embodied the knowledge in SHRDLU in pieces of executable computer code." [[Handbook vI, p295]]/.

References regarding this program include the Handbook vI, p295, Norberg and O'Neill, p316-318; McCorduck's Machines Who Think, Crevier's work; and the original dissertation.

(557) " Consider, for example, a statement of the form A implies B. As it stands, it is a simple declarative statement. But in Planner it can instead be interpreted as the imperative: set up a procedure that will see if A is ever asserted, and if this happens assert B" Minsky and Papert." 1968-1969 Progress Report." MIT AI Memo 200. p23. Also see Hewitt," Functional Abstraction in Lisp and Planner". MIT Project MAC. AI Memo 151. January 1968; and Hewitt's Ph.D. thesis, Memo 137.

(558) Minsky and Papert," Proposal to ARPA for Research on Artificial Intelligence 1970-1971." MIT AI Memo 185. December 1970, p10.

(559) Minsky CBI OH interview; personal communication, Mike Travers regarding Stallman and his foundation of the Free Software Foundation; Levy 1984.

(560) Van Atta et. al 1990, p21-6.

(561) See Levy 1984 account regarding the early development of the hacker-anarchist movement, apparently originating at MIT in the 1960s

(562) Richard Stallman," EMACS: the Extensible, Customizable Self-Documenting Display Editor." MIT AI Lab AI Memo 519A. 26 March 1981, p27; and the ARPA-IPTO Principal Investigators' Meeting, 3.12.1975, Section re MIT by P. Winston: " T. Knight and R. Greenblatt have implemented many of the important steps for a LISP machine, short of actually building the hardware, including: specification of the CONS micro processor and the LISP, the micro assembler, the basic bootstrappable LISP software."

(563) Minsky," Mini-Robot Proposal to ARPA." MIT AI Lab. Memo 251. January 1972, p13.

(564) Winston, Patrick." Proposal to ARPA". MIT AI Lab. AI Memo 366. May 1976.

(565) Campbell-Kelly and Aspray 1996, p130.

(566) Rodgers 1969.

(567) See Tom Wolfe 1982 article.

(568) Noyce 1977, p64.

(569) As Tom Wolfe recounted:

" They had decided to move into the most backward area of computer technology, which was data storage, or "memory." A computer's memory was stored in ceramic ringlets known as cores. Each ringlet contained one "bit" of

information, a "yes" or a "no," in the logic of the binary system of mathematics that computers employ. Within two years Noyce and Moore had developed the 1103 memory chip, a chip of silicon and polysilicon the size of two letters in a line of type. Each chip contained four thousand transistors, did the work of a thousand ceramic ringlets, and did it faster. The production line still consisted of rows of women sitting at tables as in the old shed-and-rafter days, but the work bays now looked like something from out of an intergalactic adventure movie. The women engraved the circuits on the silicon photographically, wearing antiseptic Mars Voyage suits, headgear, and gloves because a single speck of dust could ruin one of the miniature circuits. The circuits were so small that "miniature" no longer sounded small enough. The new word was "microminiature." Everything now took place in an air- conditioned ice cube of vinyl tiles, stainless steel, fluorescent lighting, and backlit plastic." (Wolfe 1983, p350).

(570) Reid 1984, p75.

(571) Wolfe 1983, p350.

(572) 1996, p237.

(573) One of the most interesting features of this process was its alchemy. The earliest contenders consistently offered more than they had at the time, and under pressure managed to make good on their offers. The offers turned into gold. Eric Roberts, who offered the Altair for sale, had only a kit rather than an actual a functioning machine. Gates and Allen offered Roberts the Basic for the 8088 chip before they had ever written such a language, and they sent him this offer on the stationary for the now-defunct company that they had developed as high-school students. Steve Wozniak did indeed have a machine, but until he invented the tiny tape drive that made it useful, it wasn't very useful. Roberts then devised a desperate plan for a kit for computers, sent a fake ad to Popular Electronics, and then received thousands of responses; massive pent-up demand for the Altair (Shurkin 1984, p308).

(574) This document is now in the Stanford archives.

(575) Press, Larry." Before the Altair: the History of Personal Computing", C-ACM 36, 9 (September, 1993): 27-33.

(576) 1966 formation of the amateur computer society for building computers; (Campbell-Kelly and Aspray 1996, p225.

(577) Freiburger and Swaine 1984, p2.

- (578) Englebart oral history.
- (579) Cite Engelbart oral history interview.
- (580) 16) OOP for AI p61-62.
- (581) OOP for AI p62.
- (582) Tools for Thought p205, p223.
- (583) Campbell-Kelly and Aspray 1996, p266.
- (584) Campbell-Kelly and Aspray 1996, p264.
- (585) Campbell-Kelly and Aspray 1996, p260.
- (586) Campbell-Kelly and Aspray 1996, p225.
- (587) Shurkin 1984, p315.
- (588) C/C++ for Exsys p64.
- (589) Marvin L. Minsky, Charles Babbage Institute OH Interview #179, 1989.
- (590) ARPA-IPTO Principal Investigator Meeting Agenda, January 8-10, 1973.
- (591) (2). Semantic Network Memory
" This year we have completed a prototype of a program, SCHOLAR, which is capable of conducting a mixed initiative dialogue in English with a user, based upon information stored in a semantic memory. This work has been applied to a computer-assisted instruction task conducted under government sponsorship. We are using a semantic network to represent information and combining this memory with generative techniques for originating questions and detailed program behavior. Generative techniques produce program behavior which has not been anticipated in full detail but which still follows some general guidelines. For example, generative techniques can produce computer questions related to answers (perhaps w errors) given by the student in response to prior questions... the system is readily adaptable to other tasks." BBN/ Daniel Bobrow; ARPA-IPTO Principal Investigator Meeting, Feb, 1971.
- (592) 3) ARPA-IPTO Principal Investigator Meeting Agenda, January 8-10, 1973.
- (593) The national coordination center for the ARPAnet, housed at BBN.
- (594) Ibid., statement of Frank Heart of BBN, ARPA-IPTO Principal Investigator Meeting Agenda, January 8-10, 1973; also statement regarding work done at UCSB, in the same document]]

- (595) ARPA-IPTO Principal Investigator Meeting, Feb, 1971. TOC, list of Attendees, and misc short papers;
- (596) Marvin Minsky, "A framework for representing knowledge", 1975. In Winston, ed. The Psychology of Computer Vision. New York: McGraw-Hill. 1975. Reprinted in George Luger, ed. Computation and Intelligence: Collected Readings. Cambridge, MA: The MIT Press. 1995: p163.
- (597) Patrick Winston, CBI OH interview 196, 1990.
- (598) Summarized references to scripts and frames are, respectively, the precis in the Handbook of Artificial Intelligence, vI, and in T. Winograd's thoughtful essay, "Frame Representations and the Declarative/procedural controversy" (In Bobrow and Collins, eds. 1975: 185-210). Gardener's The Mind's New Science provides a summary as well. More discursive discussions are found in Schank and Colby's Computer Models of Thought and Language (1973) and in Schank's 1980 articles in Cognitive Science ("Language and Memory") and in Bobrow and Collins' volume.
- (599) Schank 1975, p236.
- (600) Sam, Margie, and the Cyrus program, by Janet Kolodner in the mid-1970s.
- (601) A paper widely circulated as MIT AI Lab Memo 306 prior to its publication in 1975 in The Psychology of Computer Vision (a compilation of MIT papers edited by Patrick Winston
- (602) Stylianou et al C-ACM, p44.
- (603) Patrick Winston, Proposal to ARPA", MIT AI Lab. AI Memo 366, May 1976, p25
- (604) Barr interview, July 1994.
- (605) Crevier 1992, in-person interview.
- (606) References for Meta-Dendral include the Handbook vII, p117-124; the KSL's precis of its projects, Applied Artificial Intelligence Research: The Heuristic Programming Project 1990, p46, and Hayes-Roth, Waterman, and Lenat's Building Expert Systems, p7-9. And, like all mothers, it is underappreciated.
- (607) Handbook II, p120.
- (608) Hayes-Roth, Lenat, and Waterman 1982, p9-10. References to MYCIN: Buchanan and Smith 1988, p27;

Buchanan et al 1979; Handbook VII, p267; Newell's introduction to Buchanan and Shortliffe 1984, Bruce Buchanan, and Edward Feigenbaum: Buchanan and Shortliffe, 1983; Shortliffe, Buchanan and Feigenbaum, 1979; and the Handbook's articles, VII, p184-192; and Peter Karp's interesting geneology of frame and production system inferencing.

(609) Buchanan CBI interview 230, 1991.

(610) This dissertation was singularly successful. Shortly after he wrote it, Shortliffe was appointed to an assistant professorship in medical informatics at both the Medical School and the Computer Science department.

(611) Hayes-Roth, Lenat, and Waterman 1982, p9-10.

(612) Handbook II, p188.

(613) Handbook II, p188.

(614) Handbook IV, p181.

(615) Crevier 1993.

(616) References for EMYCIN are Van Melle 1979; Van Melle, Shortliffe and Buchanan 1981 (a) and (b); Buchanan 1988; Hayes-Roth Waterman and Lenat 1982, p10, and KSL Applied Artificial Intelligence Research: The Heuristic Programming Project, p47.

(617) Fagan et al., 1979.

(618) KSL Applied Artificial Intelligence Research: The Heuristic Programming Project, 1990, p47.

(619) KSL Applied Artificial Intelligence Research: The Heuristic Programming Project; p48.

(620) References for Prospector and KAS include Duda, Gaschnig Hart 1979; Building Expert Systems 1982, p10; and the Handbook II, p155-162; a general shell for constructing consultation systems is EXPERT, 1979, Weiss and Kuliowski.

(621) Hayes-Roth, Lenat, and Waterman 1982, p10 gives the exact list; also Handbook II, p155-162.

(622) McDermott 1981, 1983.

(623) Handbook IV, p181.

(624) KSL Applied Artificial Intelligence Research: The Heuristic Programming Project, p48.

References: Molgen was based upon the 1979 KSL/HPP doctoral dissertation of Peter Friedland, who will appear later as the leader of NASA's implementation of expert

systems. Friedland, P." Knowledge-based experiment design in molecular genetics". KSL Dissertation. 1979.

(625) Friedman 1991, p564.

(626) 1968; also 1972.

(627) Tello, p61-62.

(628) Wegner "25 Years article, p8.

(629) H. Penny Nii's scholarly discussion of Blackboard architecture indicates that Pandemonium was the Ur-agent and Ur-production system as well; Handbook of AI, vIV.

(630) Simon 1966.

(631) Newell et al. 1973.

(632) Handbook vI, p327.

(633) Handbook vI, p332.

(634) Handbook I, p328

(635) Handbook vI, p328.

(636) Nii, "Blackboard Systems", Handbook IV, p20.