AI Magazine Volume 4 Number 4 (1983) (© AAAI)

Introduction to the COMTEX Microfiche Editor of Memos from the Stanford Artificial Intelligence Laboratory

Bruce G. Buchanan

Department of Computer Science
Stanford University
Stanford, California 94305

THE STANFORD Artificial Intelligence Project, later known as the Stanford AI Lab or SAIL, was created by Prof. John McCarthy shortly after his arrival at Stanford in 1962. As a faculty member in the Computer Science Division of the Mathematics Department, McCarthy began supervising research in artificial intelligence and timesharing systems with a few students. From this small start, McCarthy built a large and active research organization involving many other faculty and research projects as well as his own. There is no single theme to the SAIL memos. They cannot be easily categorized because they show a diversity of interests, resulting from the diversity of investigators and projects. Nevertheless, there are some important dimensions to the research that took place in the AI Lab that I will try to put in historical context in this brief introduction.

There are far too many authors represented by SAIL reports to attempt to place each person, or report, in context. Instead, I have recounted some of the early history of SAIL, and its pre-history, as I remember it and have learned it from others' memories.[1] It is undesirable (and impossible besides) to try to unify the reports into a single theme, or to unify the research themes into a single purpose. Therefore, this mini-history mentions several themes (and a few names) from the 1960's and 70's that set the major directions of AI research at Stanford. Many of these early interests, such as robotics, have been vigorously pursued ever since. Omissions are unintentional, and should not be interpreted as having implied significance.

The present collection is a complete set of SAIL memos from the beginning of the lab until 1982.[2] The technical memos in the SAIL series are not of uniform quality. Some of these papers are preprints of journal articles of lasting interest. Others constitute documentation on how to use the system. Still others are hastily written drafts describing work in progress at the time.

Background

As mentioned, the beginnings of AI work at Stanford were modest. McCarthy had come from MIT where he had developed the LISP language and had worked on timesharing. He had published a paper in 1958 on a proposed program called the "Advice Taker"[3] which, more than any other,

[1]I thank John McCarthy, especially, for answering numerous questions and for reading the whole introduction for accuracy His advice, "don't try to unify the reports," pre-empted any contrary obligations I felt to readers Les Earnest was very helpful in giving me names and dates, providing photographs, and reading this account. I also appreciate time and information from Ed Feigenbaum, Raj Reddy, Jerry Feldman, Cordell Green, Roger Schank, Tony Hearn, Bill McKeeman and Nils Nilsson

[2]Readers should note that since the early 1970's there have been two centers of AI research at Stanford, SAIL and the Heuristic Programming Project (HPP). The HPP memos are not discussed here and are not part of the COMTEX collection.

seems to capture a definition of his own research directions in subsequent years including the present Briefly, for those not familiar with this landmark paper, the Advice Taker is a program (not implemented) that represents its knowledge about the world in formal logic and deduces the consequences of new statements about the world as soon as it is told them. In discussing this goal, McCarthy wrote:

> "Before describing the advice taker in more detail, I would like to describe more fully our motivation for proceeding in this direction. Our ultimate objective is to make programs that learn from their experience as effectively as humans do. It may not be realized how far we are presently from this objective. It is not hard to make machines learn from experience how to make simple changes in their behavior of a kind which has been anticipated by the programmer. In our opinion, a system which is to evolve intelligence of human order should have at least the following features:
>
> 1 All behaviors must be representable in the system. Therefore, the system should either be able to construct arbitrary automata or to program in some general-purpose language
>
> 2 Interesting changes in behavior must be expressible in a simple way.
>
> 3 All aspects of behavior except the most routine must bc improvable In particular, the improving mechanism should be improvable.
>
> 4. The machine must have or evolve concepts of partial success because on difficult problems decisive successes or failures come too infrequently.
>
> 5. The system must be able to create subroutines which can bc included in procedures as units. The learning of subroutines is complicated by the fact that the effect of a subroutine is not usually good or bad in itself. Therefore, the mechanism that selects subroutines should have concepts of an interesting or powerful subroutine whose application may be good under suitable conditions.
>
> Of the five points mentioned, our work concentrates mainly on the second. We base ourselves on the idea that in order for a program to be capable of learning something it must first be capable of being told it."

Over the next twenty-five years, as evidenced by many reports in this series, McCarthy's work has involved many aspects of expressing facts about the world, and changes in one's knowledge about the world, in a simple way. Common sense reasoning requires representing many facts and relations explicitly, many more implicitly. It requires rapid deductions from one simple set of statements to others, avoiding traps. There are many partial solutions to the problem of representation proposed throughout the short history of AI, much of it strongly influenced by McCarthy's hammering at the point that an intelligent program must be able to prove theorems about its own knowledge. Thus, he believes its representation of knowledge must be in a formal language.

There are many other examples of papers in this series, besides McCarthy's, of work on the problem of representation and inference. But there are other themes that were being explored within the AI Lab in the early days

The computing environment at Stanford when McCarthy arrived did not include timesharing. An IBM 650 and Burroughs 220 machine were replaced early in 1963 by an IBM 7090 and Burroughs 5000. McCarthy was given a PDP-1 by DEC and subsequently upgraded it (jointly with Prof. Patrick Suppes). It became the first display-based timesharing system after Philco delivered twelve displays in response to Stanford's specifications. In those days, McCarthy is reported to have had thoughts of a PDP-1 flying a small airplane with optical feedback from a TV camera.

DARPA support for a Stanford Artificial Intelligence Laboratory was received in 1963. In 1965 a proposal was accepted for a large increase in support permitting acquisition of a large time-sharing system and beginning work on computer vision and robotics. In 1966, the PDP-6 computer arrived and the Laboratory moved to the Donald C. Power Laboratory on Arastradero Road, about three miles from thc main campus.

Research Projects

Within about four years, there were five major team projects on the PDP-6, plus systems work and a number of individual projects. Those five were:

- Speech Understanding
- Hand-Eye
- Dendral
- Higher Mental Functions (including Natural Language Understanding)
- Computer Music

The Speech Understanding project began as Raj Reddy's PhD thesis with John McCarthy. Raj and Bill McKeeman[4] were the first two PhD's from Stanford's newly-formed (in Feb., 1965) Computer Science Department. The work was begun on the PDP-1 as an experiment in how much a computer could combine signal processing and information processing techniques. After his PhD, Raj stayed as a faculty member and leader of the Speech Project, which included Lee Erman, among others. It grew into a large and significant

[3]McCarthy, John "Programs with Common Sense," in Mechanization of Thought Processes (Proc. Symposium, National Physical Laboratory) vol 1, pp 77-84 London, Nov , 1958. Reprinted in M Minsky (ed.) Semantic Information Processing Cambridge, MA: MIT Press, 1968.

[4]McKeeman was not in AI. His thesis, under George Forsythe, was in compilers The next two PhD theses in the Computer Science Department are also in this series. Barbara Huberman Liskov worked with McCarthy on problem solving in chess, and Jim Painter worked with him on the mathematical theory of computation

project within SAIL and the greater AI community, and when Reddy and Erman moved to Carnegie-Mellon this work blossomed into the HEARSAY program.

Jerome Feldman, now at the University of Rochester, joined the AI Lab in the fall of 1966, also as a faculty member in the CSD. Jerry took over management of the Hand-Eye Project, having brought with him from MIT's Lincoln Laboratory both interest and experience with 3-dimensional graphics and representation of 3-dimensional shapes. McCarthy was interested in developing computer-controlled robot manipulators that were flexible (i.e., intelligent) in their behavior and had collected several researchers and students and experimental equipment. A prosthetic arm, built by Rancho Los Amigos in Los Angeles, was the first to operate under computer control. Affiliated faculty and students from Mechanical Engineering were also collaborating on the design of new mechanical hands and arms. Part of this work involved designing languages by which the manipulators could be programmed. And, a little later, it also involved coupling TV cameras to the computer that controls the arm to make the system responsive to changes in the environment. This was a project of many facets, as evidenced by the many different publications,but these can be categorized into three distinct aspects of the Hand-Eye work:

- mechanical engineering
- vision
- systems work

There was a long interaction with Prof. Bernard Roth and other faculty in Mechanical Engineering on the design of mechanical hands and arms. Out of this work, mainly by Victor Scheinman, for example, came the well-known PUMA arm that is now used in many industrial settings. Several persons were working on optical feedback through a TV camera. Early experiments with vision involved color and stereo vision. Considerable work on low-level vision problems, such as edge detection, by Karl Pingle, Manfred Hueckel and others, resulted in powerful capabilities. The Hueckel operator, for example, is still used in many vision systems for edge detection and, in extended form, for line following. The systems work in the Hand-Eye Project was largely driven by needs for more computational speed and more powerful languages for processing TV signals and controlling continuous movement of a manipulator. The SAIL language, initiated by Jerry Feldman before coming to Stanford, was developed in response to these needs, for example, combining data structures of LISP with an ALGOL-like syntax. Bob Sproull and Dan Swinehart contributed to this effort. Another language effort was Karl Pingle's work on a robotics language for controlling an arm. This grew into the AL language, and is the precursor of much current research on programmable robots.

Demonstrations of integrated hand-eye capabilities were made in the early 1970's. Lou Paul, for example, made a first, pseudo-demonstration showing the system stacking blocks to solve the popular "instant insanity" puzzle. A subsequent demonstration was of the arm, with TV-vision, putting together a water pump from a Model-T Ford. One of the later demonstrations involving considerably new research and engineering, was Hans Moravec's computer-controlled cart. With a TV camera mounted on top, it steered itself around the redwood deck and the road circling the lab.

Early in 1965, Edward Feigenbaum joined the Stanford faculty and began working with Prof. Joshua Lederberg, then in the Department of Genetics, on an AI project investigating scientific inference. This was the DENDRAL project, with which I worked after my arrival at Stanford in 1966. Many of the papers in this series are about DENDRAL. One of the main themes in these papers is that domain-specific knowledge – both textbook knowledge and experts' heuristics – is necessary for high performance problem solving. In the case of DENDRAL, the program uses considerable general knowledge of chemistry and specific knowledge of mass spectrometry to infer the molecular structure of chemical compounds from analytic data. It was a new dimension to work at the AI lab because of its emphasis on representing and using domain-specific knowledge of chemistry and mass spectrometry in detail instead of relying on the power of general-purpose problem solvers.

The DENDRAL project moved away from the AI Lab in the early 1970's, but continued to publish reports in this series. Gradually, the project broadened and became known as the Heuristic Programming Project with its own series of reports. This project is represented by many publications in this series. The focus of most of the DENDRAL work described in these publications is the AI (and chemistry) problems involved in interpreting analytic data about the molecular structure of chemical compounds.

Ken Colby, a psychiatrist, headed a project he called the Higher Mental Functions Project. He was then a Research Associate in Computer Science working on computer models of human belief systems and paranoid modes of thought and verbal behavior. His work on PARRY was widely known around the AI community because it was fun (and sometimes instructive) to interact with. PARRY was a vehicle for work on refining Turing's test to provide validations of simulation models. Colby's publications in this series illustrate another main theme in AI research that has been pursued more vigorously at Carnegie-Mellon than Stanford: the use of computers to model, and thereby understand, human thought.

Colby's work was also the focus of the early research on natural language understanding at SAIL. It involved mostly ELIZA-like key-word parsers for understanding dialog. Roger Schank's early conceptual grammars were done in this context, for example. (Schank became a faculty member in Linguistics, but still kept his research base at SAIL.) Among the students working with Colby and Schank, whose work is represented in the SAIL memos were Horace Enea, Larry Tesler and Bill Faught. After Terry Winograd joined the Stanford faculty, considerably more work was published in this area.

The computer music project was started when John Chowning, working with Prof. Leland Smith in the Department of Music, needed computing resources to develop his ideas on computer-based composition. They involved several of the hackers (systems programmers) from SAIL and many music students. It became one of the largest consumers of computer time (mostly at dawn, to the consternation of the students). The computer music project continued to grow and eventually inherited the building in which the ideas were nurtured.

While these projects were growing, McCarthy had been pursuing his ideas about formal reasoning, including proving the correctness of programs. Zohar Manna and David Luckham joined SAIL to work on this constellation of hard problems, Manna having come to Stanford as an Asst. Professor in 1969. Much of their work was first published as SAIL reports.

Cordell Green's thesis work along these formal lines represents the strong link between Stanford and the SRI AI Lab. Bert Raphael, at SRI, prodded Cordell to axiomatize his ideas about program description and program synthesis, and to look at Robinson's resolution theorem prover as the mechanism for reasoning. These ideas amplify the early statements about the Advice Taker quoted above, and have, in turn, been reinforced in the whole area of logic programming. Nils Nilsson also forged strong ties by choosing SAIL as his place of refuge from SRI duties to write a book on AI and theorem proving. Many other Stanford PhD students, besides Green, have found friendly advice, financial support, and computing services at SRI.

As in other AI research groups, game-playing was one focus of work on problem solving, representation and learning. Arthur Samuel joined the AI lab after his retirement from IBM and continued his research on machine learning with his checker-playing program. His presence lent a professional air to an environment that sometimes seemed to be born out of The Hobbit. Although he chose to publish little, his influence on AI has been substantial. Game playing at that time were seen as a challenging task domain in which to explore ideas on problem solving. At SAIL, McCarthy had worked on Kalah, Barbara Huberman (Liskov) worked with him on chess. Samuel continued his pioneering work on learning in the context of checkers, Reiter and Samuel pressed on the complexities of Go, and Waterman used poker for his research on learning in production systems. (His work on learning, reprinted here, launched the first issue of the first journal devoted to AI.) Dave Wilkins' recent thesis explored spatial and temporal patterns in chess. In early AI research, game playing avoided many of the complexities of real-world problems that we no longer shy away from.

Physical and Social Environment

The SAIL environment became a special part of AI research at Stanford. Different individuals saw different features, but no one would categorize it as mundane. The physical location was one important aspect, another the computing environment, another the strange social environment created by intense young people whose first love was hacking.

Les Earnest had come to the lab as Executive Officer to manage the ARPA contract late in 1965. Much of the freedom that characterized life at SAIL is due to Les sheltering the research staff from regulations and paperwork. The move to the D.C. Power building, in the rolling foothills behind campus, also created a sense of spontaneity that former staff members and visitors readily recall.

Soon after the move, new equipment began to arrive that provided an enviable research environment. 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 thereafter, six III display terminals arrived (and by 1971 Datadisc displays were in every office). A large (20 megaword) LIBRASCOPE swapping disk arrived early in 1967. For those times, the computing facilities were as convenient for researchers as could be found: timesharing service from a powerful machine, a large swapping disk, and good software support. Ed Feigenbaum has called it "the best computing environment within 2000 miles" (to which McCarthy asked "what was as good beyond 2000 miles?").

Work continued on LISP. Tony Hearn, a Stanford physicist, used LISP (originally on the 7090) for symbolic simplification of algebraic expressions (the REDUCE program). He was drawn into the growing problem of writing LISP programs that could run on different machines which still plagues us today. At the same time he was simplifying LISP, however, John Allen, Lynn Quam and others at SAIL were building more (non-exportable) features into a new dialect, called LISP 1.6. It contained a programming environment designed to aid programmers construct, edit and debug complex programs.

The PDP-6 was augmented in 1968 with a DEC KA10 processor with a 256K word core memory, and many disc drives were added for secondary storage. In 1976, SAIL installed a DEC KL10 as the main timesharing processor. The machines were nearly always saturated. Yet the work reported here was carried on at all hours of day and night whenever CPU time was available.

The social environment at SAIL was distinctive from the time the hackers (both students and non-students) installed the sauna and discovered they could sleep in the attic space above the ceiling. People were at the lab 24 hours a day. They invented gadgets to make their lives easier and more fun—a radio controlled channel selector on the TV, for example, and a coupling between the vending machine and computer to dispense snacks by charging them to personal accounts. They linked SAIL to the AP wire service in an early demonstration of the convenience of videotext. Bill Weiher programmed the CALCOMP plotter to print in many scripts and sizes, including Elvish, the fairy language from J.R.R. Tolkien. Larry Tesler and Les Earnest

created PUB to help automate document preparation, and many type fonts were soon available for the Xerox graphics printer. Gary Feldman created a recreational program that generated shaded portraits on the line printer—a precursor of the commercial line-printer portraits that flooded the country. Electronic games were an important part of the culture. Steve Russell, one of the inventors of Space War, had come with McCarthy from MIT and had moved Space War from the PDP-1 to the PDP-6 as one of the necessary programs for the new machine. Yet, for all the diversions, it was a supportive, friendly environment in which the superhackers, or "wizards" to use the term from The Hobbit, willingly helped the rest of us understand how to make the system work.

One of the near-magical qualities of the SAIL atmosphere was the sense of experimentation in which any problem was fair game. No one had been told about NP-complete problems: interesting problems just appeared to be hard or very hard. Feigenbaum used to say that he could often get students to solve very hard problems as long as he didn't tell them how difficult the more seasoned researchers had found them. McCarthy was known to assign programming the four-color map problem as a homework exercise. Systems programmers fearlessly modified the PDP-6 operating system to support the unusual peripheral devices around the lab. Dave Poole and others began designing improvements on DEC's hardware design, many of which were incorporated in the design of the KL10 processor. The physical atmosphere enhanced this sense of living in a fantasy world with only a wall of glass separating the natural beauty of the foothills from what seemed then to be 21st century technology.

The number of faculty, researchers and staff continued to grow over the next several years, as evidenced by the number of different authors of publications. Around 1979, McCarthy made a deliberate decision to return to his own research on formalization of common sense and not to invest so much time in running a large laboratory. (Coincidentally, all the faculty and staff of the Computer Science Department moved into one building on the main part of campus, and the D.C. Power building was turned over to the computer music project.) Research on automata theory and robotics is still being published in the SAIL series and the HPP and other CSD work on AI is being vigorously pursued. But AI research at Stanford lost some of its storybook flavor when the sign on the road outside the research place no longer read "Caution, Robot Vehicle."

APPENDIX:
DISSERTATIONS FROM THE STANFORD ARTIFICIAL INTELLIGENCE LABORATORY

1966		
Raj Reddy	CS[1]	*Carnegie Mellon University[2]
Stefan Persson	CS	*Stockholm School of Economics
1967		
James Painter	CS	IBM
Monte Callero	OR	*USAF Academy
1968		
Donald Kaplan	CS	Consultant, Toronto
Barbara Huberman	CS	*MIT
Donald Pieper	ME	Automatix
Donald Waterman	CS	Rand Corp.
1969		
Roger Schank	Ling.	*Yale
Pierre Vicens	CS	IRIA, France
Cordell Green	EE	Kestrel Institute
James Horning	CS	*University of Toronto
Michael Kahn	ME	Spectra-Physics Inc.
1970		
Irwin Sobel	EE	Columbia University
Michael Kelly	CS	BDM Corporation
Gilbert Falk	CS	BBN
Jay Tennenbaum	EE	Fairchild
1971		
Lynn Quam	CS	Fairchild
Robert Kling	CS	*UC Irvine
Rodney Schmidt	EE	Electromagnetic Systems Lab
Jonathan Ryder	CS	Bell Telephone Labs
1972		
Jean Cadiou	CS	International Institute for Applied Systems Analysis
Joseph Becker	CS	BBN
Gerald Agin	CS	SRI, AI Center
Lockwood Morris	CS	*Syracuse University
Richard Paul	CS	*Purdue University
Aharon Gill	EE	Israel Defense Establishment
Ruzena Bajcsy	CS	*University of Pennsylvania
1973		
Ashok Chandra	CS	IBM
Gunnar Grape	CS	Karolinska Institute
Yoram Yakimovsky	CS	Caltech Jet Propulsion Lab
Jean Vuillemin	CS	*UC Berkeley
1974		
Daniel Swinehart	CS	Xerox PARC
James Gips	CS	*Boston College
Charles Rieger	CS	Scion
Chris Riesbeck	CS	*Yale
Marsha Hannah	CS	Systems Control, Inc
Jim Low	CS	*Rochester University
Jack Buchanan	CS	*Harvard University
Neil Goldman	CS	USC-ISI
Bruce Baumgart	CS	Consultant
Ramakant Nevatia	EE	*USC
Malcolm Newey	CS	*University of Colorado, Boulder
1975		
Hanan Samet	CS	*University of Maryland
David Smith	CS	Xerox PARC
Sundaram Ganapathy	CS	*University of Michigan
Linda Hemphill	Ling.	System Control
1976		
Norihsa Suzuki	CS	Xerox PARC
Russell Taylor	CS	IBM
Randall Davis	CS	*MIT
Raphael Finkel	CS	*University of Wisconsin
Douglas Lenat	CS	*Stanford University
Robert C. Bolles	CS	SRI, AI Center
Robert Cartwright	CS	*Cornell University
1977		
Todd Wagner	CS	Intel Inc
William Faught	CS	Rand
David Barstow	CS	Schlumberger
1978		
Bruce Shimano	ME	Unimation Inc
Jerrold Ginsparg	CS	*University of Wisconsin
Scot Drysdale	CS	*Dartmouth College
1979		
Robert Filman	CS	*Indiana University
David Wilkins	CS	SRI
Elaine Kant	CS	*Carnegie-Mellon University
Brian McCune	CS	AI&DS Inc
1980		
Martin Brooks	CS	Ctrl Inst Ind. Rsch, Oslo
Donald Gennery	CS	Jet Propulsion Laboratory
Morgan Ohwovoriole	ME	NA
Hans Moravec	CS	Carnegie-Mellon University
William Scherlis	CS	Carnegie-Mellon University
1982		
Paul Martin	CS	SRI
Ron Goldman	CS	Stanford University
David Arnold	CS	Ungerman-Bass

[1] Departmental affiliation.
[2] Current or last known position.
* Denotes faculty positions