Early Artificial Intelligence Projects

A Student Perspective

Part of NSF's **Recovering MIT's AI Film History Project.**

Created at CSAIL by Heather Knight, August 2006

http://projects.csail.mit.edu/films

Contents

Main Paper

I. What is Artificial Intelligence?
II. Foundations
III. 1950's: Establishing a Field
IV. 1960's: Pursuing Machine Genius
V. 1970's: Leaping Forward
VI. 1980's: Boom and Crash
VII. Catching up to the Present

Appendix

I. The NSF Project Background
II. AI in Popular Culture
III. Related Organizations
IV. MIT Research Centers Timeline
V. Academic Research Centers Timeline
VI. Major AI Companies
VII. AI Projects Timeline
VIII. AI Papers Timeline
IX. Computation Landmarks Timeline

Bibliography

Image Credits

Early Artificial Intelligence Projects

I. What is Artificial Intelligence?

According to John McCarthy, the man that coined the term, “[Artificial Intelligence] is the science and engineering of making intelligent machines, especially intelligent computer programs” where “intelligence is the computational part of the ability to achieve goals in the world.”

An intelligent machine can be a machine that mimics the way humans think, feel, move and make decisions. It could also act in conjunction with a human to compliment and improve their ability to do those things. There are many possible ways to approach the challenge and this question has never had a static solution.

Even the name 'Artificial Intelligence' has been subject to argument, as some researchers feel it it sounds unscientific. They argue the word 'artificial' suggests lesser or fake intelligence, more like science fiction than academic research. They prefer to use terms like computational neuroscience or emphasize the particular subset of the field they like semantic logic or machine learning. Nevertheless, the term 'Artificial Intelligence' has gained popular acceptance and graces the names of various international conferences and university course offerings.

This paper does not attempt to come up with a precise characterization of the field. Instead, it examines what Artificial Intelligence has been so far by leading the reader through an admittedly non-comprehensive collection of projects and paradigms, especially at MIT and in the United States.

Unlike many fields, Artificial Intelligence has not had a linear progression and its research and breakthroughs have not grown toward an easily identified Sun. Computing, in contrast, has been noted for its exponential growth and improvement characterized by Moore's law, “the empirical observation that the complexity of integrated circuits, with respect to minimum component cost, doubles every 24 months.” The path of AI, however, more resembles the intertwining world wide web spiraling out and looping back in many directions.

Here you will find a rough chronology of some of AI's most influential projects. It is intended for both non-scientists and those ready to continue experimentation and research tomorrow. Included is a taste of who the main players have been, concepts they and their projects have explored and how goals have evolved and changed over time. Many will be surprised that some of what we now consider obvious tools like search engines, spell check and spam filters are all outcroppings of AI research.

II. Foundations

Though the term 'Artificial Intelligence' did not exist until 1956, the advances and ideas from the preceding decades evoked many of the future themes. At a time when digital computers had only just been invented, using programming to emulate human intelligence was barely a blip on the radar.

Understanding the context into which Artificial Intelligence was born helps illustrate the technological obstacles that researchers had to overcome in the search for machine intelligence as well as elucidating many of the original paths.

Beyond Number-Crunchers: Programmable Machines

The idea of machines that could not just process, but also figure out how to solve equations was seen as the first step in creating a digital system that could emulate brain processes and living behavior. What would it mean to have a machine that can figure out how to solve equations? Lets go through an example using basic algebra. In order to create a machine that could solve more complicated equations than 2+2=4, a machine needs to have a strategy for deciding how to do the multiple steps necessary in order to come up with the solution.

For example, if you told the machine, X+Y=7 and X=3, you would like the machine to deduce that 3 + Y = 7, then that Y = 7 – 3, then that 7 – 3 = 4, and finally that Y = 4. Assuming someone has already told the machine what '+', '-', and '=' mean, you would traditionally tell the machine how to solve those simple problems by defining a step-by-step procedure called a program.

As early as 1930, Vannevar Bush of MIT published a paper about a Differential Analyzer, doing just that for another class of mathematical problems. Computers had not even been invented yet, but his paper described a set of rules that would automatically solve differential equations if followed precisely.

The next major idea came in Turing's 1937 paper about any automatic programmable system, known as the Turing Machine. This concept establishes the redundant nature of making various kinds of programmable-devices, because any one could be set up such that it mimic the input-output characteristics of any other programmable machine.

Bush and Turing did not yet know how one would go about actually making that universal programmable device, but in 1949 Shannon would write a paper called “Information Theory” that set up the foundations for using digital electronics to represent information. This idea became the basis of using those big hulking machines to use symbols, like the X and Y in the example above, to execute complex operations.

Early 'Computers' were Room-Sized Calculators

Technology has improved by leaps and bounds since the start of World War II when computers were first coming into use. The first electronic computer ABC came in 1940, while the first programmable American computer was Mark I followed in 1944.

Constructed from wires, magnetic cores and vacuum tubes they were huge devices that literally filled rooms. They had about the functionality of a modern-day scientific calculator, but no monitor or keyboard. Instead, if you wanted the computer to compute the value of a calculation, you would punch buttons in sequence or feed in stacks of punch cards, and it would eventually print you back the results

A description of computing pioneer Grace Hopper's experience with a computer was representative of the kinds of problem computers were used for at the time:

> [Hopper] was commissioned a lieutenant in July 1944 and reported to the Bureau of Ordnance Computation Project at Harvard University, where she was the third person to join the research team of professor (and Naval Reserve lieutenant) Howard H. Aiken. She recalled that he greeted her with the words, "Where the hell have you been?" and pointed to his electromechanical Mark I computing machine, saying "Here, compute the coefficients of the arc tangent series by next Thursday."

Grace Hopper

> Hopper plunged in and learned to program the machine, putting together a 500-page Manual of Operations for the Automatic Sequence-Controlled Calculator in which she outlined the fundamental operating principles of computing machines. By the end of World War II in 1945, Hopper was working on the Mark II version of the machine. (Maisel)

Grace Hopper will go down in history as discovering and naming the first computer “bug” in 1945 as well as inventing the idea of a computer compiler, a device that can translate higher level programming languages into machine language that the computer knows how to execute.

The other revolutionary electronic creation of the decade was the Bell Labs transistor in 1947. A tribute to its importance according to wikipedia follows:

> The transistor is considered by many to be one of the greatest inventions in modern history, ranking in importance with the printing press, automobile and telephone. It is the key active component in practically all modern electronics.
>
> Its importance in today's society rests on its ability to be mass produced using a highly automated process (fabrication) that achieves vanishingly low per-transistor costs... The transistor's low cost, flexibility and reliability have made it an almost universal device for non-mechanical tasks, such as digital computing.

Analog Intelligence: Emulating Brain Function

Before the digital buzz caught on, many were asking themselves a question that has recently been having a resurgence in Artificial Intelligence. If we know how the brain works, why not make machines based off the same principles? While nowadays most do that by creating a programmed representation with the same resulting behavior, early researchers thought they might create non-digital devices that had also the same electronic characteristics on the way to that end. In other words, while new approaches try to represent the mind, analog approaches tried to imitate the brain itself.

From Shannon's mechanical 'mice' that could remember which path to take through a maze to get to the 'cheese' to the better known Grey Walter Turtles with wandering, home-seeking and curiosity drives that depended on its energy levels, the feedback and control heralded in Norbert Wiener's 1948 Cybernetics seemed like it might be better for AI than digital programming in the beginning. These machines relied on cleverly arranged circuits using resistors, capacitors and basic subcomponents , that automatically behave in a certain way based on sensor input or charge levels.

Unlike modern systems, who generally look to the brain for inspiration but ultimately do the actual programming using a computer, many early researchers believed we could create analog circuits which mimic the electrical behavior of the brain identically and therefore fundamentally replicate actions and intelligence.

III. 1950's: Establishing the Field

The fifties saw the growth of an AI community, experimentation with the first digital AI machines, the inaugural Dartmouth Artificial Intelligence Conference, and the creations of one of its strongest initial proponents, DARPA.

The Turing Test: An AI Legend

How can one know if a machine is intelligent? While the larger issue of defining the field is subject to debate, the most famous attempt to the answer to the intelligence question is in the Turing Test. With the way AI straddles a huge scope of approaches and fields, exploring everything from abstract theory to blue-sky and day-to-day applications, the question of how to judge progress and 'intelligence' thus becomes very difficult. Rather than get caught up in a philosophical debate, Turner suggested we look at a behavioral example of how one might indicate machine intelligence.

The actual test involves examining a transcript of a typed conversation between a person and a computer. If a third party could not tell which one was the human, the machine would then be classified as intelligent. The test was intended merely to illustrate a point, but has since ascended to the level of early legend in the AI community.

Even today, The Loebner Prize uses the Turing Test to evaluate artificial conversationalists and awards a bronze metal annually to the “most human” computer. Many former winners are available to talk to online. The organization also offers a $100,000 prize of to the program that can pass the test that has yet to be won.

Though its methodology and exclusive focus on human-style communication is contentious, one can not learn about AI without knowing what the Turing Test is. It is a common feature in any AI journal, class or conference and still serves to motivate the AI community though its literal goal is still far from being achieved.

Thinking Machine: The Logical Theorist

Early in 1956, two young CMU researchers, Al Newell and Herbert Simon implemented a working AI machine. Their 'Logical Theorist' had a built-in system that could deduce geometric proofs.

In honor of its 50 year anniversary, the story was reported in this year's Pittsburg Post-Gazette:

> “Over the Christmas holiday,” Dr. Simon famously blurted to one of his classes at Carnegie Institute of technology, “Al Newell and I invented a thinking machine...” Dr.

> Simon concentrated on developing “heuristics,” or rules of thumb, that humans use to solve geometry problems and that could be programmed into a computer, while Dr. Newell and Mr. Shaw in California, developed a programming language that could mimic human memory processes...

Their machine used symbolic reasoning to solve systems of equations, pioneering an AI methodology that involved programming knowledge and information directly into a computer.

The Dartmouth Artificial Intelligence Conference and General Problem Solver

The 1956 Dartmouth Artificial Intelligence Conference originated with a proposal submitted to the Rockefeller Foundation by McCarthy, Minsky, Fochester and Shannon suggesting a summer retreat dedicated to exploring potentials in the field whose name it coined.

It is interesting to note how relevant the seven research pillars they outlined still are:

1) Automatic Computers
2) How Can a Computer be Programmed to Use a Language
3) Neuron Nets
4) Theory of the Size of a Calculation
5) Self-Improvement
6) Abstractions
7) Randomness and Creativity.

Though they made little concrete progress that summer, it marked the start of an new age and McCarthy's use of the controversial name 'Artificial Intelligence' stuck.

Given that it was the first working implementation of digital AI, it might seem curious that the Logical Theorist project did not seem significantly impress the other people at the Dartmouth Conference. One explanation is that Newell and Simon had been invited to the conference almost as an afterthought, less well known than many of the other attendees.

By 1957, the same duo created a new machine called the General Problem Solver that they heralded as an epoch landmark in intelligent machines, anticipating that it could solve any problem given a suitable description.

While its ability to solve complex problems disappointed, the GPS explored and formalized the problem-solving process, and helped researchers better understand the issues at stake in achieving an effective program. It was also the first program that aimed at a general problem-solving framework and inspired much further research.

Optimism about the rate of AI Progress: GPS and NP-hard Problems

In retrospect, other established researchers admit that in they mostly followed other seemingly promising routes that did not end up working as well as the Newell-Simon approach in the decade following the Dartmouth conference. Later they acknowledged some of Newell and Simon's original insights and many joined the symbolic reasoning fold (McCorduck).

This reaction fits into a reputation that this field has of unrealistic predictions of the future. Unfortunately, many see AI as a big disappointment, despite the many ways its advances have now become a fundamental part of modern life. If you look at the rash claims of its original proponents, however, such a conclusion may not seem far fetched.

A particularly exuberant example of this disconnection was Newell's claim after the creation of General Problem Solver (GPS) that “there are now in the world machines that think, that learn and create. Moreover, ...in a visible future – the range of problems they can handle will be coextensive with the range to which the human mind has been applied.” (Norvig)

One limitation he overlooked was the curse of 'NP-hard' problems. In these cases, it is not that one can not write an appropriate program to find a solution, but that it will effectively never return an answer because the computation will take so long. That is, a fundamental property of its formulation is that execution time grows exponentially with the size of the input. In other worlds, given two inputs, the output might take 2^2 = 4 seconds to compute, three inputs might take 2^3=16 seconds, eight might take 2^8=256 seconds and so forth.

Modern researchers seem to have a more cautious approach to speculations about the future, having learned from their history. Some see AI research as a way to appreciate and understand the complexity of the human mind. It has certainly been much harder than most realized to achieve even a small part of what organic brains can do. When asked what advice they would give a novice AI researcher, one AAAI Fellow recommended, “Choose a easy problem. Then make it simpler. It will always turn out to be much harder than you'd expect.”

ARPA: Early AI's Fairy God Mother

If the Turing Test was the spirit-leader of early AI research, ARPA was the day-job that paid the bills, although one of its original heads, J. C. R. Licklider, did also encouraged many new conceptualizations of the purpose and potential of technology. Licklider's paper, *Man Machine Symbiosis,* outlined a way of envisioning the human-technology relationship, in which a machine assists and works with a human to accomplish tasks. The extensive resources that the organization provided were indispensable to the start of the field.

Short for the Advanced Research Program Association, and a subset of the Defense Department, ARPA (now known as DARPA) was created in 1958 after Sputnik I went into orbit with the explicit purpose of catching up with the Russian space capabilities. When Eisenhower decided that space should be civilian-controlled and founded NASA, however, ARPA found computing its new niche.

It began operations by contributing large research block grants starting in 1963 and supported a range of AI and computer science efforts over the years, predominantly developed (at least initially) by MIT, Stanford and Carnegie Mellon.

LISP: The language that made AI possible

John McCarthy introduced LISP in 1958, heralded as the language that made AI programming possible. LISP is special because it was the first language that allowed information to be stored as list of objects rather than just lists of numbers. An object is essentially a placeholder or symbol that is defined somewhere else. This structuring makes it possible to program recursive functions and abstract ideas directly into the machine.

John McCarthy 1967

As part of the shift of batch-processing to interactive computers, McCarthy designed LISP to have an interactive environment, in which one has the capability of evaluating and seeing on screen feedback one function at time, rather than having to run the entire file. This can greatly facilitate finding bugs in one's code.

While many other early languages have died out, LISP remains the most common programming language for Artificial Intelligence in the United States and is used on par with Prolog in Europe and Japan. According to Peter Norvig, founder of Google and author of a popular textbook on the subject, one reason for the continuing popularity of Lisp is the flexibility of its simple list data structure. In his words, “The list is a very versatile data structure, and while lists can be implemented in any language, Lisp makes it easy to use them. Many AI applications involve lists of constantly changing size, making fixed-length data structures like vectors harder to use.” (Norvig 25)

It is also easily extensible because there are no limitations on how one defines and manipulates both programs and data, so one can easily extend the language to better fit the problem at hand. Its simple elegance has survived the test of time while capturing all the necessary functionality; functions, data structures and a way to put them together.

Research at MIT: The Artificial Intelligence Project

The first coordinated AI research at MIT began in 1959 when John McCarthy and Marvin Minsky founded the Artificial Intelligence Project as part of the Research Laboratory for Electronics (RLE) in Building 26 (not Tech Square) and also the Computation Center. They were junior faculty at the time and had known each other as from graduate school at Princeton, where Minsky had studied artificial neural networks (cybernetics). A theoretician, he immediately begin work on theories of computations relevant to creating intelligent machines in *Computation: Finite and Infinite Machines.*

AI and computation have long had mutually inspiring relationship. Much AI research could not be implemented until we had different or better machines, and their theories influenced the way those strides forward would be achieved. The early gurus were often pioneer's in both fields, creators and consumers of the new technologies. The tools they created often become part of the expected package for the next generation of computers, and explored and expounded upon improvements that any new machine might have.

MIT Hackers: Starting a Computer Culture

Among others, computers captured the imagination of the formerly relays and wiring obsessed sect of the Tech Model Railroad, creating a breed of 'hackers' that believed in the power, beauty and freedom of computing. The 'Hacker Ethic' that still exists at MIT today found its roots in the fifties and, as taken from Steven Levy's book about the subject, consisted of the following precepts:

1) Access to computers – and anything which might teach you something about the way the world works – should be unlimited and total. Always yield to the Hands-On Imperative.
2) All information should be free.
3) Mistrust Authority – Promote Decentralization.
4) Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position.
5) You can create art and beauty on a computer.
6) Computers can change your life for the better.

A scant few years before, computers had only existed as a heavily regulated industry or military luxury that took up whole rooms guarded by designated personnel who were the only ones actually allowed to touch the machine. Programmers were far removed from the machine and would pass their punch card programs on to the appropriate personnel, who would add them to the queue waiting to be processed. The results would get back to the programmers eventually as a binary printout, which was then deciphered to find the result.

Thus, the Hacker's desire to play with the machine itself was revolutionary for the time. With the reverence surrounding the expensive machines, the concept of spending one's day in front of a computer at the modern office would have sounded ludicrous. In contrast to and immune to the social mores of the time, the hackers felt challenged and inspired by the worlds of possibility they saw in these new machines and the power those new machines gave them to create new virtual universes.

Hacker Innovations

In the late fifties and later, computers were on and put to work day and night because they were so expensive (and slow). So it was not uncommon for these young computer enthusiasts to keep late hours and take advantage of the less-utilized middle of the night machine time. Not only would they sign up for their own slots, but they even developed a system whereby someone would watch out for when another sleepy user did not show up for their time. The information would be immediately relayed to the rest of the group at the Model Railroad club and someone would make sure the time did not go to waste.

PDP-1 with Teletype

One of the most important hacker innovations was hooking up a screen and teletype machine to the computer, first used for interactive debugging. In doing so, users had an interactive real time relationship and drastically changed the way a user would use and relate to the machine. Some of these innovations would grow into the life, gas, and solar corona video clips available on this website.

As a result of using the machine so much, they knew where they wanted optimize machine performance and what tools to create to elicit new kinds of functionality from the machines. Early hackers created better languages and even hardwired new commands into the computer circuitry. The most famous program, however, was Space Wars, the first real computer game, involving maneuvering spacecrafts and torpedos that was created on a machine little memory and virtually no features. Soon it spread through the entire computing community, even used by the Digital Equipment Corporation to ensure the customer properly working computers. As told on wikipedia, "Spacewar was a fairly good overall diagnostic of the PDP-1 computer and Type 30 Precision CRT Display, so DEC apparently used it for factory testing and shipped PDP-1 computers to customers with the Spacewar program already loaded into the core memory; this enabled field testing as when the PDP was fully set up, the field representative could simultaneously relax and do a final test of the PDP."

IV. 1960's: Pursuing Machine Genius

In terms of projects, the sixties saw the creation of the first comprehensive mathematics programs, the first attempts at decoding sentence meaning in the case of word problems, and the creation of now integral operating system tools like user faces and word processors. A conversing parody of a psychoanalyst gained notoriety, the first industrial robot made its appearance and the expert system DENDRAL derived conclusions in the area of chemistry. If this section seems like something of a laundry list, that is because there are so many different subareas which saw their beginnings in these seminal projects.

As years progressed, each new computer would form a new image in the strobe light morphing from big hulking machine to interactive personal computer. The growing capabilities opened up new possibilities for AI. For example, imagine having a computer without a screen. It was Lincoln Labs' computer LINC that incorporated a TV-style CRT screen into a commercial computer, giving a user immediate feedback instead of making the user wait for a printout. Everything from graphics to word processing to user interfaces has hinged on that addition.

On the other coast at the Stanford Research Institute (SRI), Doug Englebart invented the mouse and on-screen cursor in his experiments with different kinds of user faces, as well as windows and multiple raster monitors, all of which he demoed in 1967.

The operating computer systems were far from failsafe. In 1960, one Defense computer mistakenly identified the moon for an incoming missile which understandably caused great consternation at the time. Another example came during the Cuban Missile crisis, when communications were blocked for several days. These shortcomings would help motivate high-level encouragement and support for the computer industry.

At the same time. computer science was gaining growing acceptance as a field. First IBM declared separate departments for software and hardware, meaning pure programmers officially would have a declared place to develop programs and environments. Then, in the academic sphere, University of Pennsylvania granted its first PhD in Computer Science to Mexelblat in 1968 and there is some discussion about whether Stanford or MIT should be credited with awarding one even before that, though it might have been under a different label.

The decade also saw the birth of the BASIC programming language, designed to be easy to understand, and UNIX, a way of structuring and communicating with an operating system that now underlays all Macs and Linux-based computers.

With the new DARPA funding in 1963, MIT created a new research group Project MAC. Mirroring the wide range of research it would inspire, Project MAC brought together disparate researchers from departments across the institute, including those from the AI Project. All moved over to Tech Square, originally occuping two floors, complete with machine shop, research areas and Minsky's beanbags and project testing haven, the play-pen.

Playing Chess, 1968

The lab, under Bob Fano's initial leadership, focused on mimicking higher cognitive levels of human intelligence. They worked on systems that could play chess, do SAT analogy problems, higher level math, and infer logical conclusions from a given set of preconditions. One fun invention was Ivan Sutherland Virtual Reality head-mounted display, the first of its kind.

Math Programs at MIT: SAINT, MACSYMA, STUDENT (ANALOGY)
Slagle, Moses, Bobrow, Evans MIT

The initial use of programs to solve complex mathematics was not a matter of rote application of straightforward computations, but rather involved programs that could actively figure out what that solution or a close approximation might be.

The first step at MIT, SAINT, was created by PhD student James Slagle and could solve basic integrations. It also had the dual fame of being the first LISP program ever written. CSAIL has a reading room that preserves the collection of all these early thesis projects, and although not the only institution that could claim this, early titles read much like a timeline of developments in AI and Computer Science at that time.

Expanding upon the more traditional approach of using computers as high-powered calculators, the mammoth MACSYMA entered the scene in 1967. The predecessor of Matlab, and still widely used by mathematicians and scientists, this program used symbolic reasoning for integration problems, logic based system.

It became the go-to system for mathematical operations and one of the earliest expert systems. Its creator was Joel Moses of MIT and he initially used a collection of mostly unstructured LISP functions to accomplish a wide variety of operations.

The different way to approach math on a computer was Danny Bobrow's thesis in 1964 that could solve high-school level algebra word problems, mixing in semantic rules for interpreting natural language. The year before, Thomas Evans had created ANALOGY, a program that could solve SAT-level analogy problems. Though not a math problem itself, ANALOGY used a way of deciphering relationships between words similar to that in Bobrow's project. Though they may seem at first glance more human, Norvig comments that these kinds of programs "derive simplicity because they deal with simplified worlds."

Building Tools at MIT: TECO, SKETCHPAD
Greenblatt and Murphy, Sutherland, MIT

TECO was a text editor created at MIT by Greenblatt and Murphy in 1962. Predominantly used for writing code at the time, the concept would evolve into the word processor functionality that would later help computers break into the workplace. In one colorful description, author Steven Levy declared the young Greenblatt a "single-minded, unkempt, prolific, and canonical MIT hacker who went into the night phase so often that he zorched his academic career."

The next big tool was SKETCHPAD, a drawing program that invented the graphical user interface. According to wikipedia, an open encyclopedia that all can edit:

> Ivan Sutherland demonstrated... that computer graphics could be utilized for both artistic and technical purposes in addition to showing a novel method of human-computer interaction.
>
> Sketchpad was the first program ever to utilize a complete graphical user interface. Sketchpad used an x-y point plotter display as well as the then recently invented light pen. The clever way the program organized its geometric data pioneered the use of "objects" and "instances" in computing and pointed forward to object oriented programming.

LOGO, 1967: early AI language.
Papert, MIT

There is a large presence of LOGO and LOGO turtle videos in the TechSquare film clips. Invented by Seymour Papert of MIT, LOGO is famous for being an easier-to-understand programming language. It pioneered the idea of educational children programming programs, the first of which occurred down the street from MIT in Lexington, MA.

Students and researchers could type in the human-friendly commands over teletype, a typewriter-like contraption that was wired into the main computer and could make simple math, word or whatever they could imagine programs.

The next major innovation came when they hooked the system up to a 'turtle' robot whose movements were scripted by the LOGO programs. It provided a way for the students and researchers to immediately see their program in action and test out their algorithms by watching its motion.

LOGO Turtle

By strapping a marker or pencil to the turtles and initiating some simple rules for movements, the robots became famous for tracing complex and beautiful patterns on the paper beneath it. Use the same algorithms to create a path in pixels and they created some of the first screensaver-like graphics.

Vision Project, 1966: They thought they could Solve Machine Vision in a Summer

By connecting cameras to the computers, researchers experimented with ways of using AI to interpret and extract information about vision data. No one really understood how difficult that would be and the initial MIT attempt is one of my favorite AI anecdotes.

Rumor has it that the task of figuring out how to extract objects and features from video camera data as originally tossed to a part-time undergraduate student researcher to figure out in a few short months. What is known for certain is that there was summer vision project sometime in the sixties, in which researchers fully expected to establish many of the main concepts by the start of the next semester.

As would often be the case in AI, they had vastly underestimated the complexity of human systems, and the field is still working on how too make fully functional vision systems today.

UNIMATE, 1961: The First Industrial Robot
Engelberger and Devol, General Motors

According to the Computer History Museum, "The first industrial robot UNIMATE started out in 1961 on the TV picture tube manufacturing line, then went to work at General Motors. Weighing 4,000-pounds, the robot arm that obeyed commands one-by-one to stack and sequence die-cast metal."

Robots would become a major area in AI experimentation, with initial applications in factories or human controllers but later expanding into some cooperative and autonomous tasks as well. The world 'robot' is derived from the Czech word for worker, but nowadays the machines are used from everything from actors in the Entertainment Industry (see the movies Gremlins, A.I.) to high precision medical surgeries, toys and autonomous vacuum cleaners.

ELIZA, 1965: A pattern-matching psychologist
Weizenbaum, MIT

ELIZA is a simple pattern matching conversational machine, the first of her kind, created by MIT computer scientist Joseph Weizenbaum in 1966. It has now become part and parcel of all Emacs programs and crudely simulates a psychotherapist using pattern matching.

Weizenbaum was not a proponent of AI. In fact, in the paper about his invention, he says, "Machines are made to behave in wondrous ways, often sufficient to dazzle even the most experienced observer. But once a particular program is unmasked, once its inner workings are explained in language sufficiently plain to induce understanding, its magic crumbles away; it stands revealed as a mere collection of procedures, each quite comprehensible."

Whether or not you agree with Weizenbaum's characterization, the best way to understand the program is to look at an example. The following comes from the description Daniel Bobrow gave of an unplanned encounter between the Vice President of BBN an abandoned ELIZA program left running on the computer screen:

> Here is how Bobrow tells the story:
>
> One Saturday morning about 9 am., a Vice President of Bolt, Beranek, and Newman [BBN]... found a note on the PDP-1 teletype that read: "TYPE ALL MESSAGES INTO THIS TELETYPE AND END THEM WITH A PERIOD." This note was left there by one of the programming bums who had just succeeded in making run a LISP version of the DOCTOR program (based on ELIZA as developed by Weizenbaum). However, the VP thought that this note indicated that this teletype was linked with the one in my home, since I often use the computer all night. Therefore, he started typing to "me," but in fact was typing to this version of the DOCTOR program. The following conversation ensued:
>
> **vp:** Do you think that I can use the computer this morning?
>
> **eliza:** Why do you ask?
>
> **vp:** I might be able to make some additional Telcom sales.
>
> **eliza:** Why aren't you so sure?
>
> **vp:** My perspective customers have never seen a demonstration of the system.
>
> **eliza:** Does this mean very much to you?
>
> **vp:** Of course it does.
>
> **eliza:** You seem very positive about that.
>
> **vp:** This is ridiculous.
>
> **eliza:** What do you mean, this is ridiculous?
>
> **vp:** Is there any reason why I can't use the computer now?
>
> **eliza:** Tell me more about your perspective customers.
>
> **vp:** Please dial me up on 491-1850
>
> Note that after the last remark the VP did not obey instructions, and left out the period. Therefore, of course, the computer didn't answer him. This so infuriated the VP, who thought I was playing games with him, that he called me up, woke me from a deep sleep, and said:
>
> **vp:** Why are you being so snotty to me?
>
> **bobrow:** What do you mean I am being snotty to you?
>
> The VP angrily read me the dialog that "we" had been having, and couldn't get any response but laughter from me. It took a while to convince him it really was a computer.

The basic algorithm is the following: (1) Look at user's input, (2) Find a pattern that matches the input, (3) Look up the and print out the corresponding response. Though you can, of course, form your own opinion, I find it amazing that such a simple setup can result in such an amusing and complex situation.

DENDRAL, 1966: Chemistry Expert System analyzing organic compounds
Buchanan, Feigenbaum, Lederberg, Sutherland, Stanford

One of the clearest examples of applied AI research, DENDRAL analyzed organic compounds using mass spectrogram and nuclear magnetic resonance data to determine their structure. It limited the search space using constraint satisfaction, increasing the probability that the system would find a solution.

The heuristics and rules it used to trace the path of which structures and characteristics respond to what kind of molecules were painstaking gathered from interviewing and shadowing experts in the field. It involved a very different approach to intelligence from a universal problem solving structure, requiring extensive specialized knowledge about a system.

It evolved into the MetaDendral system, which attempted to automate the knowledge gathering bottleneck of building an expert system, and which made the first scientific discovery by a machine in 1975.

V. 1970's – A Rising Industry

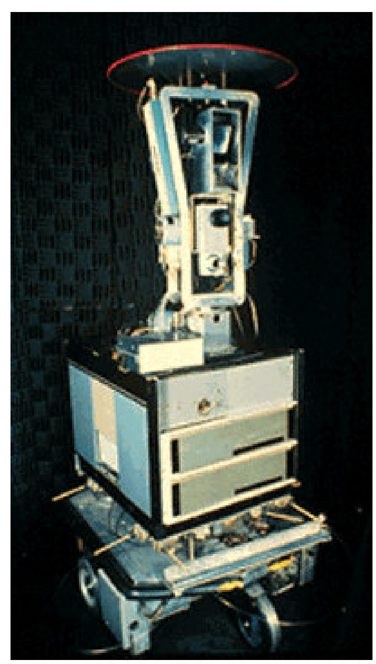
Directions of AI advancement accelerated with the introduction of the first personal computers, a medical diagnostic tool MYCIN, new conceptualizations of logic, and games like Pong and PacMan.

Expanding from tools to applications, Project Gutenburg began compiling electronic versions of books in 1970, an ongoing effort now available online. The first reading machine was created by Kurzweil in 1976 and was used to assist the blind. Whether robots or keyboards, the next evolutionary step in both AI and computer science came with the control, interpretation and coordination of peripheral devices.

Computers, inaccessible to individuals outside of military, academia and large banks, were suddenly conceivable to own oneself for a mere few thousand dollars. At the start, the machine didn't even have a screen, just a set of LEDs and buttons to punch in sequence in order to program the machine. But it was a start and market forces soon welcomed in a flood of peripheral devices to better input and output. As Microsoft and Apple Computers began operations and the first children's computer camp happened in 1977, major social shifts in the status of computer technology were underway.

Back at MIT, former director Rod Brooks relates that in the seventies, "Patrick Winston became the director of the Artificial Intelligence Project, which had newly splintered off Project MAC. The lab continued to create new tools and technologies as Tom Knight, Richard Greenblatt and others developed bit-mapped displays, fleshed out how to actually implement time-sharing and included e-mail capabilities.

"Knowledge representation, knowledge-based systems, reasoning and natural language processing continued to motivate innovations in projects programming languages as the lab expanded in size, accepting former students Gerry Sussman, Carl Hewitt and Ira Goldstein into the faculty ranks."



Shakey, 1968

Early Mobile Robots: Shakey, Freddie
Stanford and University of Edinburgh

DARPA funded various initial robot projects across the country including Stanford's mobile robot Shakey. In a similar vein, the University of Edinburgh soon created their own mobile robot, Freddie in 1973. Both robots used visual perception and other inputs to create internal models of the world around it, which they would then use to navigate through space. More specifically, wikipedia declares:

> SRI International´s Shakey became the first mobile robot controlled by artificial intelligence. Equipped with sensing devices and driven by a problem-solving program called STRIPS, the robot found its way around the halls of SRI by applying information about its environment to a route. Shakey used a TV camera, laser range finder, and bump sensors to collect data, which it then transmitted to a DEC PDP-10 and PDP-15. The computer radioed back commands to Shakey — who then moved at a speed of 2 meters per hour.

Robots and Natural Language Processing at MIT: Copy Demo, Silver Arm, SHRDLU
MIT

Early robotics included the 1961 MH1 robot-hand project and 1970 copy-demo in which a robotic arm equipped and camera could visually determine the structure of a stack of cubes and then construct an imitation. Both of the projects are well documented on the AI Films page.

Cube Stacking Experiments

The first first computer controlled robot intended for small parts assembly came in 1974 in the form of David Silver's arm, created to do small parts assembly. It's fine movements and high precision required great mechanical engineering skill, as well as using feedback from touch and pressure sensors. Patrick Winston soon expanded on the idea of cube manipulation with his program ARCH, that learned concepts from examples in the world of children's blocks.

Another major program applied to the blocks world was SHRDLU, a natural language processing created by Terry Winograd in 1970. According to wikipedia:

> SHRDLU is a program for understanding natural language, written by Terry Winograd at the M.I.T. Artificial Intelligence Laboratory in 1968-70. SHRDLU carried on a simple dialog (via teletype) with a user, about a small world of objects (the BLOCKS world) shown on an early display screen (DEC-340 attached to a PDP-6 computer).
>
> The name SHRDLU was derived from ETAOIN SHRDLU, the arrangement of the alpha keys on a Linotype machine, arranged in descending order of usage frequency in English.
>
> A side effect of this memory, and the original rules SHRDLU was supplied with, is that the program could answer questions about what was possible in the world and what was not. For instance, SHRDLU would deduce that blocks could be stacked by looking for

> examples, but would realize that triangles couldn't be stacked, after having tried it. The "world" contained basic physics to make blocks fall over, independent of the language parser.
>
> Finally, SHRDLU could also remember names given to objects, or arrangements of them. For instance one could say "a steeple is a small triangle on top of a tall rectangle"; SHRDLU could then answer questions about steeples in the blocks world, and build new ones.

To clarify, an example SHRDLU dialog might consist of the following:

> Person: PICK UP A BIG RED BLOCK.
> Computer: OK.
> Person: GRASP THE PYRAMID.
> Computer: I DON'T UNDERSTAND WHICH PYRAMID YOU MEAN.
> Person (changing their mind): FIND A BLOCK WHICH IS TALLER THAN THE ONE YOU ARE HOLDING AND PUT IT INTO THE BOX.
> Computer: BY "IT", I ASSUME YOU MEAN THE BLOCK WHICH IS TALLER THAN THE ONE I AM HOLDING.
> Computer: OK.
> Person: WHAT DOES THE BOX CONTAIN?
> Computer: THE BLUE PYRAMID AND THE BLUE BLOCK.
> Person: WHAT IS THE PYRAMID SUPPORTED BY?
> Computer: THE BOX.
> Person: HOW MANY BLOCKS ARE NOT IN THE BOX?
> Computer: FOUR OF THEM.
> Person: IS AT LEAST ONE OF THEM NARROWER THAN THE ONE WHICH I TOLD YOU TO PICK UP?
> Computer: YES, THE RED CUBE. (wikipedia)

The SHRDLU research was exciting because it allowed the user, albeit in highly constrained circumstances, to communicate directly with the computer in English, rather than having to learn a machine programming language.

MYCIN: Medical Diagnostic Expert System
Edward Shortliffe, Stanford

MYCIN was the first fully developed expert system, diagnosing and recommending treatment for infectious blood diseases. The name 'Mycin' was used because it is a common antibiotic suffix. Created in the beginning of the decade, by 1979 MYCIN was declared as good as medical experts by the Journal of American Medical Association. The system even adjusted recommended dosages to the patient's body weight. It's functionality can be described as follows:

> Mycin operated using a fairly simple inference engine, and a knowledge base of ~500 rules. It would query the physician running the program via a long series of simple yes/no or textual questions. At the end, it provided a list of possible culprit bacteria ranked from high to low based on the probability of each diagnosis, its confidence in each diagnosis' probability, the reasoning behind each diagnosis (that is, Mycin would also list the questions and rules which led it to rank a diagnosis a particular way), and its recommended course of drug treatment.
>
> Mycin was never actually used in practice. This wasn't because of any weakness in its performance — in tests it outperformed members of the Stanford medical school. It was as much because of ethical and legal issues related to the use of computers in medicine

> — if it gives the wrong diagnosis, who can be held responsible? Issues with whether human experts would find it acceptable to use arose as well. (wikipedia)

The creators of MYCIN found that doctors were unwilling to accept its advice if the system could not convince them of why it made its conclusions. Therefore, they included the ability to answer questions about how it was making its decisions. As described in one AI textbook, “[MYCIN] uses rules that tell it such things as “If the organism has the following set of characteristics as determined by the lab results, then it is likely that it is organism X.”

By reasoning backward using such rules, the program can answer questions like “Why should I perform that test you just asked for?” with such answers as “Because it would help to determine whether organism X is present.” (Rich 59) It is important that programs provide justification of their reasoning process in order to be accepted for the performance of important tasks.

VI. 1980's: Boom and Crash

The start of the eighties was the golden age for Artificial Intelligence in the US, as the field caught the imagination of the larger population. Institutions across the board were suddenly springing up departments of Artificial Intelligence from video game companies to Campbell's Soup. The most common utilities came in the form of MYCIN-style expert systems, wizards that could give advice or information about how to do something it its area of expertise.

These expert systems were specialized system meant to preserve the knowledge base of gurus in the field. For example, in the case of Campbell's soup, a factory manager might be curious about the tub-cleaning requirements between making different batches of soup. If you were going from Chicken Broth to Chicken Noodle, you could proceed right way, but if the ordering was Clam Chowder to Vegetarian Minestrone, the tanks better be spic and span in between.

Family and work computers started to become commonplace in the 1980's with six million computers sold in 1983. Most of the tool builders at MIT left the lab in the eighties to work in new companies and bring their work to the consumer. IBM had introduced its 'PC' two years earlier and Xerox, LMI and Symbolics had also introduced a variety of Lisp machines. Apple's LISA and then Macintosh hit the market and ARPANET, which would open the way for the Internet opened up to civilians. By the end of the decade, however, the 'AI Winter' left the field struggling to defend its funding and reputation. On campus, however, research continued unabated in both old and new directions.

In 1985, Professor Nicholas Negroponte and former MIT President Jerome Wiesner started the MIT Media Laboratory. According to the Media Lab website:

> [The Media Lab grew] out of the work of MIT's Architecture Machine Group, and building on the seminal work of faculty members in a range of other disciplines from cognition and learning to electronic music and holography... In its first decade, much of the Laboratory's activity centered around abstracting electronic content from its traditional physical representations, helping to create now-familiar areas such as digital video and multimedia. The success of this agenda is now leading to a growing focus on how electronic information overlaps with the everyday physical world. The Laboratory pioneered collaboration between academia and industry, and provides a unique environment to explore basic research and applications, without regard to traditional

divisions among disciplines. (MIT Media Lab webpage)

The MIT AI lab was also in full swing, directing its talents at replicating the visual and mobility capabilities of a young child, including face recognition, object manipulation and the ability to walk and navigate through a room. Tomas Lozano-Perez pioneered path search methods used for planning the movement of a robotic vehicle or arm. There was work done on legged robots by Marc Raibert and John Hollerback and Ken Salisbury created dexterous robot hands. Famed roboticist Rodney Brooks also built his first robots.

Wabot-2, 1980: Robot that reads Sheet Music and plays Organ

Waseda University, Japan

The name WABOT is from 'WAseda roBOT', honoring the University in Japan at which it was designed. In this case, the story is best told by its originators. The description of the project as told on the Waseda University website follows:

> It has been forecast that robots will penetrate society in 21st century... In that case, robots will be required to have anthropomorphic appearance sand faculties... Developing the anthropomorphic intelligent robot WABOT (WAseda roBOT) [aimed] to finally develop a "personal robot" which resembled a person as much as possible.
>
> In 1980, our laboratories... commenced the WABOT-2 project. Playing a keyboard instrument was set up as an intelligent task that the WABOT-2 aimed to accomplish, since an artistic activity such as playing a keyboard instrument would require human-like intelligence and dexterity.
>
> ...The robot musician WABOT-2 can converse with a person, read a normal musical score with is eye and play tunes of average difficulty on an electronic organ. The WABOT-2 is also able of accompanying a person while he listens to the person singing. The WABOT-2 was the first milestone in developing a "personal robot." (Waseda 2006)

WABOT playing music

It is interesting to note that the research group sees WABOT-2 as the first generation of an oncoming class of personal robots. It may seem far-fetched at the moment, but look how far personal computers have come since they were first conceived of fifty years ago.

HEARSAY, 1982: Speech Understanding Program

Erman, Hayes-Roth, Lesser, Reddy at CMU

HEARSAY was a speech understanding program developed at CMU in 1982 that pioneered a useful model for solving perceptual problems, that is, problems in which a machine is trying to derive meaning out of complex input signals. That process might involve decoding words from someone's voice, recognizing someone's face from a set of vision data or tactilely distinguishing different kinds of textures.

Because it is a widely applicable problem, below you will find a textbook summary of the steps one must consider in figuring out how a machine can glean information from sensory data. As HEARSAY was it is a CMU project, it seems appropriate to include a summary from the an Artificial Intelligence textbook by Elaine Rich of CMU:

> **TECHNIQUES USED IN SOLVING PERCEPTUAL PROBLEMS...**
>
> It is important to divide the overall understanding process into manageable pieces. We can do this by dividing the process of analyzing either a speech sample or a picture into the following five stages:
> **Digitization:** Divide the continuous input into discrete chunks. For speech recognition, this can be done by measuring the amplitude of the signal at fixed intervals, such as 20,000 times per second...
> **Smoothing:** Eliminate sporadic large variations in the input. Because the real world is mostly continuous, these spikes in the input are usually the result of random noise.
> **Segmentation:** Group the small chunks produced by digitization into larger chunks corresponding to logical components of the signal. For speech understanding, these segments correspond to logical components of the signal... such as *s* or *a*. These segments are often called *phones*...
> **Labeling:** Attach to each of the segments a label that indicates which, of a set of building blocks, that segment represents... So the labeling procedure can do one of two things. It can assign multiple labels to a segment and leave it up to the later analysis procedure or choose the one that makes sense in the context of the entire input. Or it can apply its own analysis procedure in which many segments are examined to constrain the choice of label for each segment.
> **Analysis:** Put all the labeled segments together to form a coherent object... when surrounding pieces are considered, the number of interpretations that lead to a consistent overall interpretation [also known as constraint satisfaction] is considerably reduced.. In speech, this results from such things as intonation patterns that cover whole sentences. (Rich 349)

The actual HEARSAY program pulled this information together using a blackboard model that uses this techniques in a way that traces up and down the complexity levels as well as right to left on sentences when more ambiguous signals come through. Like constructing a jig saw puzzle, one first identifies the easily recognizable, and then fill in the less obvious pieces in between. This becomes particularly useful when words are not enunciated clearly.

Aaron, 1985: An Autonomous Artist
Harold Cohen, UCSD

Harold Cohen with AARON

Harold Cohen an English artist who almost accidentally encountered programming at Stanford, was father to the first robot artist in his creation, AARON. Who knows what the rising technological art community will come up with next. According to Cohen's homepage:

> The AARON program, an ongoing research effort in autonomous machine (art making) intelligence... began when [Cohen] was a visiting scholar at Stanford University's Artificial Intelligence Lab in 1973. Together, Cohen and AARON have exhibited at London's Tate Gallery, the

Brooklyn Museum, the San Francisco Museum of Modern Art, Amsterdam's Stedelijk Museum and many more of the world's major art spaces...

One of the few artists ever to have become deeply involved in artificial intelligence, Cohen has given invited papers on his work at major international conferences on AI, computer graphics and art technologies...

> AARON has produced many thousands of drawings, to a few dozen of which Cohen has added color... The painting machine with which AARON colored real drawings in the real world was premiered at an exhibit at the Computer Museum in Boston in the spring of 1999." (Homepage)

A picture being created by the latest version of AARON side by side with its creator appears above.

Allen, 1985: Starting a New Generation of Reactive Robots
Rodney Brooks, MIT AI Lab

One of the original AI Lab groups was named the Mobot Lab and dedicated to making mobile robots. Allen was the group's first creation and shares Brook's middle name.

According to author Kevin Kelly:

> "Allen" was the first robot Brooks built. It kept its brains on a nearby desktop, because that's what all robot makers did at the time... The multiple cables leading to the brain box from Allen's bodily senses of video, sonar, and tactile were a never ending source of frustration for Brooks and crew... Brooks vowed that on their next project they would incorporate the brains inside a robot -- where no significant wiring would be needed -- no matter how tiny the brains might have to be.
>
> They were thus forced to use very primitive logic steps, and very short and primitive connections in "Tom" and "Jerry," the next two robots they built. But to their amazement they found that the dumb way their onboard neural circuit was organized worked far better than a brain in getting simple things done.

Since then, Rodney Brooks has become one of the most famous proponents of robotics and is the current head of CSAIL, MIT's Computer Science and Artificial Intelligence Laboratory.

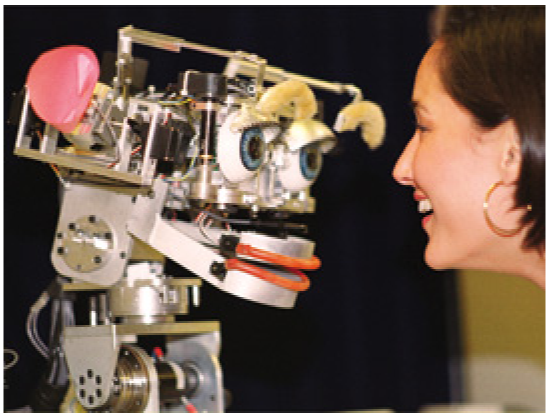
VII. Catching up to the Present


RoboCup 2006

Since the eighties, several projects stand out as major new shifts and developments in the field. When Deep Blue beat world chess champion Garry Kaspacov in 1996, some say it marked the end of an era in which specialized programs and machines reigned. One new potential direction, the first official RoboCup, kicked off that the very same year posing and requires integrating all kinds of intelligences. Their goal is to be able to beat the winning World Cup soccer team by 2050.

With the results of the DARPA Grand Challenge this year, that potentially rash aspiration seems like it could actually be plausible. After the first year's race when none of the autonomous vehicles made it even ten miles past the start of the 131.2 mile course, this year saw five of the twenty-three DARPA Grand Challenge competitors reach the finish with hours to spare.

In 2002 efforts started to recreate a once wonder of the world status library in Egypt in an online e-book version called Bibliotheca Alexandrina. The transition to computerized medical records has been sluggish, but in other areas of medicine from imagery to high precision surgery, the new facilitates machines can give a surgeon has saved lives and made new diagnosis and operations possible.



Cynthia Breazeal with Kismet

While we have all heard about NASA space robots, but less known were the $400,000 'His' and 'Her' robots featured in the 2003 Niemen Marcus Christmas catalog. Clearly, our relationships with machines in society is in transition. One of the most important examples of that was Cynthia Breazeal's research on machine emotion and social interaction with her MIT thesis-project Kismet in 2002.

New versions of ELIZA-like programs are becoming commonplace with AOL Instant Messenger's SmarterChild, an agent that can answer questions and try to search the web to answer your questions about Movie times or tell you not to have a 'potty mouth.'

While we do not have full realization of Licklider's man-machine symbiosis, the idea of machines and tools becoming agents that work hand and hand with human beings seems more and more natural with each generation. IRobot's vacuum cleaner Roomba is kickstarting a new household robotics industry with record sales. John McCarthy believes that fundamental new ideas are required before AI can reach human-level intelligence, rather than just needing large databases and faster computers. He declares on his website, “My own opinion is that the computers of 30 years ago were fast enough if only we knew how to program them.”

Whether or not human-level intelligence is even the main goal of the field anymore, it is one of the many that entice our interest and imagination. It is clear that AI will continue to impact and contribute to a range of applications and only time will tell which paths it will travel along the way.

Heather Knight received her B.S. in Electrical Engineering with a minor in Mechanical Engineering in 2006, having done research at the Media Lab since 2002 with Professor Cynthia Breazeal of the Robotic Life as well as Professor Deb Roy of Cognitive Machines and has been accepted into the MIT EECS Masters of Engineering program.

Appendix

I. Project Background

The *Recovering MIT's AI Film History* project was born in 2001, when a collection of old film reels showed up on some dusty shelves during the move from Tech Square to Frank Ghery's artitectural creation, the Ray and Maria Stata Center. The Stata Center is the home of the now joined AI Lab and Computer Science departments known as CSAIL, the Computer Science and Artificial Intelligence Laboratory.

Thanks to the support of the National Science Foundation, these films and more are now available on the project website, **http://projects.mit.edu/films**. The original NSF proposal to digitize and create a website was worded as followed:

> This project will collect, organize and preserve historic materials, particularly film, that are part of the historical record of the field of Artificial Intelligence (AI). It will create an organized digital archive and use highlights selected from the archive to illustrate the intellectual history of AI... Sources for this project included notes, memos and technical reports from MIT and elsewhere, and in particular, a uncatalogued, unconserved and uncurated collection of films that recently came to light at MIT... The project will create a web site or DVD to showcase the selected clips, the connecting narrative, and other more technical materials.

Lucy and the Minsky-Bennett Arm

The opening of the website fortuitously coincided with both the 50th anniversary of AI, as the term was coined at the Dartmouth conference in 1956, and the American Association of Artificial Intelligence (AAAI) conference in Boston, MA June 16-22, 2006.

There we had the opportunity to interview on video more than one quarter of the AAAI Fellows in attendance, and have now included them on the sight. The Fellows include the most influential innovators in the field of Artificial Intelligence as deemed by AAAI and many of the original founders were present.

Another primary source for the site was Rick Greenblatt, who began his MIT career in the 1960s. He was extraordinarily generous with his time, watching each and every of the site's film clips and leaving an audio 'podcast' of his reminiscences for each one.

The Recovering MIT's AI Film History website itself was created over the summer of 2006, led by

CSAIL's Outreach Officer Tom Greene and produced by Luis Gomez (University of Florida undergrad), Heather Knight (MIT MEng student) and Matt Peddie (MIT undergrad), who collectively did the research, web design and interviews contained within the site.

I would like to personally thank MIT Electrical Engineering and Computer Science Professors Fernando Corbato and Bob Fano, as well Harvard History of Science PhD candidate Hallam Stevens for reading drafts of this paper. I have not done full justice to the feedback they offered, but the content is more complete and somewhat less error-ridden after their help.

II. Artificial Intelligence in Popular Culture

Asimov, Isaac. I, Robot (1950), Caves of Steel (1954), Robots of Dawn(1982). Robot Science Fiction, book. Conceives fictional Three Laws of Robotics
Orwell, George. 1984 (1949). Big Brother uses computers to enslave humanity, book.
Shelley, Mary Frankenstein. book.
Kubrick, Stanley. “2001: A Space Odyssey” (1968), movie. (Based on book by Arthur C. Clark)
“Star Wars” (1977), movie.

III. AI Organization Timeline

1951 IEEE founded.
1956 The Dartmouth AI Conference, McCarthy coins name.
1958 DARPA created.
1958 Teddington (UK) Conference. McCarthy, Minsky, Selfridge
1969 First IJCAI Conference in Washington DC.
1974 First SIGGRAPH conference.
1980 First AAAI conference. Stanford.
1982 ICOT formed. Japan.

IV. MIT Research Centers Timeline

1959 Artificial Intelligence Project starts, led by Minsky and McCarthy
1963 Project MAC, led by Minsky and Papert
1969 AI Lab splits off from Project MAC, led by Pat Winston
1975 LSC (Laboratory of Computer Science) replaces Project MAC
1980 The Media Lab founded by Negropante?
2003 CSAIL (Computer Science and Artificial Intelligence Laboratory) grows out of a LCS and AI Lab merger, co-directed by the former heads of both, Victor Zhu and Rod Brooks prospectively.

V. Academic Research Centers Timeline

1959* MIT's Artificial Intelligence Project, founded by John McCarthy and Marvin Minsky.
1963 Stanford AI Lab (SAIL), founded by John McCarthy
1963 MIT's Project MAC, begun under Minsky and Seymour Papert, $2 million DARPA grant.
CMU AI Lab, founded

1966 Edinburg AI Lab, founded by Donald Michie.
1979 CMU Robotics Institute, founded by Raj Reddy.
1980 MIT Media Laboratory

VI. Major AI Companies:

(very incomplete)

DEC, Artificial Intelligence Corp., Apple, Microsoft, Symbolics, Xerox, Intel, LMI, Teknowledge, Thinking Machines, Google

VII. AI Projects Timeline

1947 Grey Walter builds electro-mechanical "turtle"
1949 Turing and colleagues try to create a chess program on Mach 1.
1950 Chess Program proposed as search problem. Shannon.
1956 The Logic Theorist, solves math problems. Newell, Shaw and Simon.
1957 General Problem Solver, "means-end analysis." Newell, Shaw and Simon.
1959 Checkers Program beats best human players. Samuel.
1959 Timesharing. Kurtz and Kemeny.

1961* SAINT, first Lisp program. PhD work. J. Slagle.
1962* TECO, text editor for PDP-1. Murphy and Greenblatt. MIT.
1962 First Commercial Industrial Robots
1963* ANALOGY, solves SAT-level analogy problems. PhD work. Thomas Evans.
1963* SKETCHPAD, drawing tool. Sutherland.
1963 Parser, tested on "Time flies like an arrow." Susumo.
1964* STUDENT, solves high-school level algebra word problems. PhD. Danny Bobrow.
1964* SIR. PhD work. Bert Raphael.
1965* ELIZA, conversational psychotherapist. Joseph Weizenbaum.
1965* First Virtual Reality head-mounted display. Ivan Sutherland.
1966 DENDRAL, chemistry knowledge-based sys. Buchanan, Feigenbaum, Lederberg, Sutherland. Stanford.
1967* LOGO, early AI language. Papert.
1967* MACSYMA, symbolic reasoning for integration problems, logic based system. Joel Moses.
1968* Tentacle Arm, aka Minsky-Bennett arm.

1970 PROLOG. Alain Colmerauer.
1970 Shakey, first computer controlled mobile robot. Stanford.
1970 INTERNIST, aid in disease diagnosis. Pople and Myers.
1970* SHRDLU, natural language processing, blocks world. Terry Winograd.
1970* ARCH. Winston.
1970 Project Gutenburg, free electronic versions of books. M. Hart.
1971 PARRY, paranoid conversation agent. Colby.
1971 STRIPS, first motion planning system?. Nils Nilsson and Rich Fikes.
1972 Smalltalk. Xerox Parc.

1972 PONG, early video game. Nolan Bushell.
1973 Scripts developed. Schank and Abelson.
1973 MYCIN, medical diagnostic expert system. PhD Edward Shortliffe. Stanford.
1974* Silver Arm, first computer controlled robot, intended for small parts assembly. David Silver.
1975 MetaDendral, first scientific discovery by a machine.
1976 Adventure, first adventure game. Crowther and Woods.
1976* First LISP machine. Greenblatt.
1976 First reading machine. Kurzweil.
1976 Automated Mathematician. Lenat.
1976* Primal Sketch for Visual Representation. David Marr et al.
1979 Stanford Cart crosses chair filled room without help. Hans Moravec.
1978 VisiCalc. Bricklin.
1978 Version Spaces. Tom Mitchell. Stanford.
1978 MYCIN generalized. PhD. Bill VanMelle. Stanford.
1979 PacMan brought to market.

1980 HEARSAY, uses blackboard model. Erman, Hayes-Roth, Lesser, Reddy. CMU.
1980 Expert systems up to 1000 rules.
1980 Japanese 5th Generation Project. Kazuhiro Fuchi.
1981 Connection Machine Designed, powerful parallel architecture. Danny Hillis. Thinking Machines.
1983 SOAR. John Laird & Paul Rosenbloom with Allen Newell. PhDs. CMU.
1984 Neural Nets with backpropagation widely used. John Hopsfield.
1984 "Wabot-2" reads sheet music and plays organ.
1985 Aaron, autonomous drawing program. Harold Cohen.
1985* Allen, autonomous reactive robot. Rod Brooks.

1990 Human Genome Project begins
1997 Deep Blue beats world chess champion Garry Kaspacov?.
1997 First Official RoboCup, start of a new paradigm
2000 Kismet, robot that recognizes and displays emotion. PhD. Cynthia Breazeal.
2000 AIBO introduced.
2002 Bibliotheca Alexandrina
2003 Niemen Marcus's Christmas catalog features $400,000 his and her robots.

VIII. AI Papers Timeline

1930* Differential Analyzer, Vannevar Bush, MIT
1937 "On Computable Numbers," Turing Machine. Turing.

1943 Neural Networks. McCulloch and Pitts.
1945* "As We May Think." Vannevar Bush, MIT.
1948 "Cybernetics." Norbert Wiener.
1949 Information Theory. Shannon.

1950 "Computing Machinery and Intelligence," Turing Test. Turing.

1957* “Syntactic Structures.” Chomsky.
1958* Perceptron, Rosenblatt.

1962 “Structure of Scientific Revolutions.” Kuhn.
1962 “Possible Worlds Semantics.” Kripke.
1963 Semantic Networks as a Knowledge Representation. M. Ross Quillian.
1963* “Steps Toward Artificial Intelligence.” Marvin Minsky.
1968* “Semantic Information Processing.” Marvin Minsky.
1968 *“The Sound Pattern of English.” Chomsky and Halle.
1969* “Perceptrons,” discusses limits of single layer neural networks. Minsky and Papert.
1969* “Philosophical Problems from the Perspective of Artificial Intelligence,” situation calculus McCarthy and Pat Hayes.

1972 “What Computers Can't Do.” Dreyfus.
1974* “A Framework for Representing Knowledge.” Marvin Minsky.
1974 “Creative Computing.” Ahl.
1974 “Computer Lib.” Nelson
1976 Metalevel reasoning, PhD. R. Davis. Stanford.
1979 Mycin as good as medical experts. Journal of American Medical Association.
1979* AI Lab Flavors OOP memo. Weinreb and Moon.
1979* Non-monotonic logics. McDermott and Doyle (MIT), McCarthy (Stanford).

1980 “The Knowledge Level.” Allen Newell.
1980 “Gödel, Esher, Bach,” wins Pulitzer. Hofstadter.
1983 “The Fifth Generation.” Feigenbaum and McCorduck.
1984 “Common LISP the language.” Steele.
1985* “The Society of Mind.” Marvin Minsky.

IX. Landmarks in Computation

1940 The ABC, first electronic computer. Atanasoff and Berry.
1941 Z3, first programmable computer. Zuse. Germany.
1944 Mark I, first programmable computer in US. Aiken.
1945 First computer “bug.” Grace Hopper.
1947 Transistor. Schockley, Brittain and Ardeen. Bell Labs.

1950 UNIVAC, first commercial computer. Eckert and Mauchley.
1952 Compiler. Grace Hopper.
1956 FORTRAN, programming language. IBM.
1958 Integrated Circuit. Jack St. Clair Kilby.
1959 PDP-1 sells for $159,000. DEC.

1960 Defense computer mistakes moon for incoming missile.
1960 LINC, first computer with integrated CRT. Lincoln Labs.
1961 All Dartmouth students required to be computer literate. Kemeny's timesharing system.

1964 PDP-8, first mass-produced microcomputer. DEC.
1964 IBM 360 series.
1964 BASIC, programming language. Kemeny and Kurtz.
1967 IBM distinguishes hardware and software.
1967 Mouse, windows and multiple raster monitors demoed. Englebart. SRI.
1968 First PhD in Computer Science. Mexelblat. University of Pennsylvania.
1969 UNIX, Thomson and Ritchie. AT&T.

1970 Floppy Disks.
1971 Intel 8008, first microprocessor in US.
1975 BASIC for a microcomputer, Gates and Allen.
1975 Altair 8800, first personal computer with 256 bytes memory.
1975 BYTE, magazine.
1977 Apple Computer. Wozniak and Jobs.
1977 Apple II, Radio Shack TRS80, Commodore PET.
1977 First children's computer camp.
1977 Microsoft founded.

1980 Lisp machines widely marketed. Xerox, LMI, Symbolics.
1981 IBM Introduces Personal Computer (PC)
1983 Six million computers sold.
1984 Apple LISA
1984 Compact Disk (CD) technology. Sony.
1984 Apple introduces Macintosh.
1987 ARPANET opens to civilians

Bibliography

"Artificial Intelligence: The Next Twenty-Five Years." Edited by Matthew Stone and Haym Hirsh. AI Magazine, 25th Anniversary Issue. Winter 2005.

Brooks, Rodney. "Artificial Intelligence Laboratory." Electrons and Bits. Ed. John V. Guttag. Cambridge, MA, Electrical Engineering and Computer Science Department: 2005.

Buchanan, Bruce and McCarthy, John. AAAI 2002. Brief History of Artificial Intelligence. <http://www.aaai.org/AITopics/bbhist.html>.

Buchanan, Bruce G. "A (Very) Brief History of Artificial Intelligence." AI Magazine, 25th Anniversary Issue. Winter 2005.

Chandler, David. Volkswagen wins robotic race across the desert. NewScientist.com news service. Oct. 10, 2005 <http://www.newscientist.com/article.ns?id=dn8119>.

Cohen, Paul R. "If Not Turing's Test, Then What?" AI Magazine, 25th Anniversary Issue. Winter 2005.

Edwards, Paul N. Closed World: Computers and the Politics of Discourse in Cold World America. Cambridge, MA: The MIT Press, 1996.

Garfinkel, Simon L. LCS: Architects of the Information Society. Ed. Hal Abelson. Thirty-Five Years of the Laboratory for Computer Science at MIT. Cambridge, MA: The MIT Press, 1999.

Greenblatt, Rick. "Podcasts." Recovering MIT's AI Film History Website. MIT. June 2006. <http://projects.csail.mit.edu/films>.

Güzeldere, Güven, and Stefano Franchi. "Dialogues with Colorful Personalities of early AI." SEHR: Constructions of the Mind. Vol. 4.2, 24 July 1995. <http://www.stanford.edu/group/SHR/4-2/text/toc.html>.

"Harold Cohen." Personal Homepage at Center for Research in Computing and the Arts. University of California San Diego. 1999 <http://crca.ucsd.edu/~hcohen/>.

"Harold Cohen's 'Aaron' – The Robot as an Artist." SciNetPhotos. 1997 <http://www.scinetphotos.com/aaron.html>.

Kelly, Kevin. "Machines with Attitude" Out of Control: The New Biology of Machines, Social Systems and the Economic World. Chapter 3. Perseus Books Group: 1995 <http://www.kk.org/outofcontrol/ch3-b.html>.

Kirsh, David. "Foundations of artificial intelligence: The big issues." Artificial Intelligence 47 (1991): 3-30.

Levy, Steven. Hackers. New York: Dell Publishing Co., 1984.

Luger, George. “AI: Early History and Applications” Ch1 of Artificial Intelligence: Structures and Strategies for Complex Problem-Solving. Addison Wesley; 4th edition. January 15, 2002. <http://www.cs.unm.edu/%7Eluger/ai-final/chapter1.html>

MIT Project Oxygen. MIT, 2006. <http://oxygenlcs.mit.edu/Overview.html>

MIT Media Laboratory. MIT, 2006. <http://www.media.mit.edu>.

Maisel, Merry and Laura Smart. “Admiral Grace Murray Hopper.” Women in Science. San Diego Supercomputer Center, 1997 <http://www.sdsc.edu/ScienceWomen/hopper.html>.

McCarthy, John. “Reminiscences on the History of Time Sharing.” Stanford University, 1983. <http://www-formal.stanford.edu/jmc/history/timesharing/timesharing.html> 2006.

McCarthy, John, M.L. Minsky, N. Rochester, C.E. Shannon. “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence.” August 31, 1955. <http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html>.

McCarthy, John. “What is Artifical Intelligence?” Website FAQ. 24 Nov. 2004. <http://www-formal.stanford.edu/jmc/whatisai>.

McCorduck, Pamela. Machines who Think. (Original ed. 1974). Natick, MA: A K Peters, 2004.

Minsky, Marvin. “Steps toward Artificial Intelligence.” Computers and Thought. Ed. Edward Feigenbaum. place: publisher, 1963: 406-450.

Nilsson, Nils J. “Human-Level Artificial Intelligence? Be Serious!” AI Magazine, 25th Anniversary Issue. Winter 2005.

Norvig, Peter. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp. San Francisco, CA: Morgan Kaufman Publishers, 1992.

Turing, A.M. “Computing Machinery and Intelligence” Computers and Thought. * 1963: 11-35.

Rich, Elaine. Artificial Intelligence: International Student Edition. The University of Texas at Austin. Singapore: McGraw-Hill, 1983.

“Robots and AI Timeline.” The Computer History Museum. Mountain View, CA. 2006 <http://www.computerhistory.org/timeline/timeline.php?timeline_category=rai>.

Spice, Byron. “Over the holidays 50 years ago, two scientists hatched artificial intelligence.” Pittsburg Post-Gazette. 2 Jan. 2006. <http://www.post-gazette.com/pg/06002/631149.stm>.

“WABOT: Waseda roBOT.” Humanoid Robotics Institute. Waseda University, Japan.

<http://www.humanoid.waseda.ac.jp/booklet/kato02.html>.

Waldrop, M. Mitchell. The Dream Machine: J.C.R. Licklider and the Revolution That Made Computing Personal. New York: Penguin, 2002.

Various Articles. Wikipedia, August 2006. <http://wikipedia.org>.

Photo Credits

1. Graphic (Project Website) <http://projects.csail.mit.edu/films/pictures/Screenshots/24-worm2.jpg>

2. Grace Hopper (San Diego Supercomputer Center) <http://www.sdsc.edu/ScienceWomen/hopper.html>

3. John McCarthy (Computer History Museum) <http://www.computerhistory.org/about/press_relations/chess/>

4. PDP-1 (Computer History Museum) <http://www.computerhistory.org/timeline/timeline.php?timeline_category=rai>

5. Chess (Project Website) <http://projects.csail.mit.edu/films/pictures/Screenshots/13-cubechess2.jpg>

6. LOGO Turtle (Project Website) <http://projects.csail.mit.edu/films/pictures/Screenshots/22-turtle.jpg>

7. Shakey (Computer History Museum) <http://www.computerhistory.org/timeline/timeline.php?timeline_category=rai>

8. Cube Stacking (Project Website) <http://projects.csail.mit.edu/films/pictures/Screenshots/13-cubechess.jpg>

9. WABOT (Waseda University) <http://www.humanoid.waseda.ac.jp/booklet/kato02.html>

10. AARON (SciNetPhotos) <http://www.scinetphotos.com/aaron.html>

11. RoboCup (RoboCup Website) <http://www.robocup2006.org/sixcms/detail.php?id=117&lang=en>

12. Kismet (MIT Media Lab) <http://web.media.mit.edu/~cynthiab/NewFiles/research.html>

13. Lucy on Minsky-Bennett Arm (Project Website) <http://projects.csail.mit.edu/films/pictures/Screenshots/61-arm.jpg>