John McCarthy Awards for Excellence in Research and Research Environments
November 22, 2009

1:30-2:30 PM: NEW SAIL PRESENTATIONS Conducted by Sebastian Thrun

Mike Montemerlo for leadership and technical contributions that led to the DARPA Grand Challenge victory.
Michael Montemerlo is being recognized for his leadership in Stanford Autonomous Driving Team, which led to the victory in the DARPA Grand Challenge. Mike was the chief software architect of the system, and his work influenced all aspects of Stanley's software, from hardware interfaces, perception, mapping, path planning, and control. Mike was the technical leader for the entire team, and his vision and technical strength guided all other team members in this successful project. The DARPA Grand Challenge was widely considered a milestone event for robotics. For the first time, robotic cars proved their ability to navigate extensive desert trails completely autonomously. Mike's contributions to this project were immeasurable, for which he now receives this award.

Hendrik Dahlkamp for control algorithms that led to the DARPA Grand Challenge victory.
For his contributions to Stanford's autonomous vehicle Stanley, and specifically its computer vision system for long-range road detection. Called Stanford's "secret weapon" by a PBS documentary on the DARPA Grand Challenge, this system enabled Stanley to perceive and classify desert terrain in the distance as drivable or undrivable, and determine a safe traversal speed. The algorithm was a crucial contribution in two ways: First, it allowed Stanley to extend its sensing range from a classical laser-based perception range of 20 meters to a camera-based range of 40 meters, which led to a 40% increase in top speed and a win in the race. Second, it advanced a new paradigm in artificial intelligence called "self-supervised learning", where the output of one sensor modality, the laser range finder, is used to generate online training data for a second sensor modality, the camera. This allowed Stanley to constantly adjust its road model to the environment, taking time-of-day, surface material, texture, shadows etc into account. Together with Intel Computer Vision Researcher Adrian Kaehler, Hendrik implemented a perception system that was able to react even to obstacles such as tank traps that it had never encountered before.

David Stavens for computer vision algorithms that led to the DARPA Grand Challenge victory.
For his groundbreaking contributions to the winning DARPA Grand Challenge vehicle, and in particular his algorithms for adaptive speed control. His adaptive speed control algorithm selected the best speed autonomously with machine learning, considering features such as road roughness, slope, and width. The algorithm could be trained to closely match a human driver's speed choices. This allowed Stanford to avoid massive human-tweaking of the race route, known as "pre-planning," that characterized some other teams. The algorithm includes a band-pass filter, designed by Gabe Hoffmann, to isolate the vehicle's suspension. David made contributions to several other aspects of the robot, including the watchdog program for software health

monitoring and the adaptive vision system. In addition, David was on the launch team responsible for the vehicle on the morning of the race and, as TA for the Grand Challenge class, oversaw the very first end-to-end development of Stanley. He was also the principal safety driver for in-the-desert road tests. His other responsibilities included team coordination and media and investor relations. Using data from the event, David later published a self-taught learning algorithm that used haptic feedback from the IMU to enhance the laser perception beyond race performance.

Pieter Abbeel for acrobatic flight maneuvers of the Stanford autonomous helicopter.
Adam Coates for acrobatic flight maneuvers of the Stanford autonomous helicopter.
Autonomous helicopter flight is widely regarded to be a highly challenging control problem. It is particularly difficult to design controllers for non-stationary maneuvers in which the helicopter goes through various flight regimes, extensively exposing the great complexity of helicopter dynamics. Despite these challenges, human experts can reliably fly helicopters through a wide range of maneuvers, including aerobatic maneuvers at the edge of the helicopter's capabilities.

Pieter Abbeel and Adam Coates developed apprenticeship learning algorithms that leverage expert demonstrations to efficiently learn good controllers for the tasks being demonstrated by an expert. These apprenticeship learning algorithms have enabled their helicopters to significantly extend the state of the art in autonomous helicopter flight and aerobatics. Their experimental results included the first autonomous execution of a wide range of maneuvers, including flips, rolls, loops, auto-rotation landings, chaos and tic-tocs, which only exceptional human pilots can perform. Their results also included complete air shows, which required autonomous transitions between many of these maneuvers. Their system performs as well, and often even better, than an expert human pilot.

Ken Salisbury for the design and build-up of the Salisbury robotic hand.
Ken Salisbury designed the "Salisbury Hand" (originally known as the Stanford/JPL hand) while he was a graduate student at Stanford as an advisee of Prof. Bernie Roth in Mechanical Engineering. In collaboration with Carl Ruoff at NASA/Jet Propulsion Labs and Prof. Roth, the hand was designed to be a platform for investigation of robotic grasping and dexterous manipulation. Commercialized through
Ken's "Salisbury Robotics" company in the early 80's this hand became a popular research platform for many years and continues to be an icon symbolic of robotic dexterity. In the years since, Ken's labs have spawned an number of well-known robotic and haptic devices, including the MIT-WAM arm (now known as the Barrett Arm from Barrett Technology), the PHANToM Haptic Interface from Sensable Technology, telesurgical devices commercialized by Intuitive Surgical, and most recently the first version of the Personal Robot that is now being commercialized by Willow Garage.

Dan Klein for revolutionary contributions to unsupervised probabilistic language parsing.
Dan Klein wins this award for his pioneering contributions to the unsupervised learning of natural language structure. Klein's thesis work demonstrated the first computer system capable of acquiring high-quality grammars from raw text alone, answering a long-

standing open question about the empirical learnability of human languages. Along with his group at UC Berkeley, he has since continued to advance the state of the art in natural language processing using unsupervised and latent-variable methods. In addition to constructing fast and accurate syntactic analysis systems, his recent research has successfully tackled a variety of other language tasks. In the area of machine translation, for example, his work on syntactic correspondence has produced the best systems for learning latent translation alignments. In the area of reference resolution, his research has led to a fully unsupervised system that outperforms its supervised competitors. Recent results on historical reconstruction have demonstrated the most accurate system for the automatic inference of ancestral words from modern forms. Klein is the recipient of multiple academic honors, most recently including the ACM Grace Murray Hopper award, a Microsoft New Faculty Fellowship, a Sloan Fellowship, and multiple best paper awards.

2:30-3:00 PM: Intermission for group photo, refreshments and video “Hear! Here!”

3:00-4:30 PM: OLD SAIL PRESENTATIONS Conducted by Raj Reddy

Bruce Baumgart for creating the SAILDART computer archive.
Preserving digital records and making them accessible for the long term is a difficult task both because digital recordings, especially those on magnetic media, don't last long and because write/read technologies keep changing as do file formats. Bruce Baumgart, with help from Martin Frost and others, has been able to preserve most of the records of the Stanford Artificial Intelligence Lab from the 1970s and '80s and has made the public files from that period publicly accessible on http://www.saildart.org. Private files are accessible there to their owners via logins. He did this with a great deal of personal effort and at his own expense. The problem of preserving such records for the very long term, as we believe they should be, is not yet solved but this effort constitutes a big step in the right direction.

Bruce Buchanan for pioneering contributions to knowledge based systems.
As a Research Associate for the DENDRAL Project, Bruce Buchanan used the SAIL time-sharing system in his pioneering work on knowledge acquisition from experts, and knowledge representation for the DENDRAL experiments. That and subsequent contributions in knowledge-based systems, and in machine learning, led to his Research Professorship at Stanford; his University Professorship at the University of Pittsburgh; his election to the National Academy's Institute of Medicine; and the Presidency of the AAAI.

John Chowning for creating the computer music synthesis system
John Chowning initiated the computer music project at SAIL, with the indispensable help of undergrad student David Poole, that became the internationally recognized Center for Computer Research in Music and Acoustics (CCRMA). Along the way he discovered a frequency modulation scheme that could closely emulated the sounds of known musical instruments and many that are unknown. Software was not patentable at that time so he recruited grad student Andy Moorer to translate it into a hardware design that was patented through Stanford and licensed to Yamaha, which eventually incorporated it into

a wide range of digital synthesis devices, from organs to cell phones, including the most widely sold synthesizer ever, the DX 7. The proceeds from that licensing agreement helped CCRMA get through a critical phase of its development—including the acquisition of the Foonly F2, built by David Poole—and eventually endow the ongoing programs at CCRMA.

Whitfield Diffie for initiating the public key cryptography development
Whitfield Diffie originated the important idea of public key cryptography, which then inspired others, such as SAIL alumnus Ron Rivest to further develop this idea. The creation of practical public key encryption systems has had major effects on protecting personal privacy and moving away from the idea that only governmental agencies have the right to encrypt and protect their records.

Les Earnest for helping to start ARPAnet and creating the social networking program FINGER
During 1967-68 Les Earnest was a member of the ten person startup committee for the first packet switching network, which came to be called ARPAnet and later turned into the Internet. Their initial performance specifications were not too foresighted in that only two functions were specified: file transfer and remote computing, which came to be called Telnet. However the file transfer capability was adequate to support email when it came into use a short time later and the tight round trip communication requirement needed for Telnet made possible the much later interactive web services.

In the 1970s Les created the FINGER program, which could show who was currently logged in and, if not, when they last logged out. This was to help keep track of SAIL people who worked at all hours of the day and night. Given that nearly all SAIL software was made publicly accessible, a number of other laboratories with similar computer systems took copies of FINGER for their own use and soon requested that a network version be developed that could check on people at other sites, which Mark Crispin developed. FINGER also allowed each person to create a Plan file, tied to their email address, to describe such things as their planned work schedule or vacation plans. However in short order FINGER became a de facto social networking system, given that it facilitated finding which of one's friends were online and allowed people to post what amounted to personal blogs some 30 years before the term "blog" came into use. For more see http://asia.cnet.com/reviews/pcperipherals/0,39051168,61998604,00.htm .A Unix version was created by a UC Berkeley group that unfortunately had a security loophole that was exploited by the first Internet Worm, resulting in FINGER being suppressed on security grounds. Google is now developing a modern version called WEBFINGER.

Ralph Gorin for creating the first spelling corrector
The first spelling checker was created at MIT in 1961 by Les Earnest as part of the first cursive handwriting recognizer and used a list of the 10,000 most common English words. In 1967 Earnest recruited a SAIL graduate student to make a spelling checker for text files, which was written in LISP, used a suffix stripping scheme to effectively increase the vocabulary of the word list, and rather slowly produced a list of unrecognized words and their locations in the file. In 1971 Earnest recruited Ralph Gorin to make an interactive spelling checker. Gorin wrote SPELL in machine language, for

faster action and made the first spelling corrector by searching the word list for plausible correct spellings that differ by a single letter or adjacent letter transpositions. The program became more useful by allowing each user to extend the dictionary interactively and use those extensions in the future. He made SPELL publicly accessible and it soon spread around the world via the new ARPAnet, about ten years before personal computers came into general use.

Anthony Hearn for creating the Standard Lisp System
After the Lisp programming language was created by John McCarthy and his colleagues at MIT during the late 1950s, it became the most widely used language in artificial intelligence research because of its versatility and extensibility. However its extensibility became a problem as many different versions began to appear which were incompatible with each other. After Tony Hearn began developing a symbolic computation system called REDUCE, he addressed this problem by creating and documenting Standard Lisp in an attempt to bring the diverging branches back together. This idea was later picked up by others to create Common Lisp.

Victor Scheinman for developing high performance robot arms
A computer should be able to do physical work. A big computer should be able to work fast. As a Mechanical Engineering grad student, Vic was engaged to design and build a series of robot arms and other gadgets for the PDP computers to play with. With Bernie Roth, Larry Leifer, Don Pieper, Mike Kahn, Lou Paul, Bruce Shimano and others we learned from his pneumatic snake like digital arm (the ORM-1966), and a powerful and fast hydraulic arm (1967) which ran in “spacewar mode” and shook the building that the design of the robot needed to be compatible with the brain and it’s environment. His electric “Stanford Arm” (1969) became the Hand-Eye group standard research robot manipulator for many years. He built several of these arms for other research groups including General Motors, National Bureau of Standards (now NIST), and AT&T. He also had a strong hand in the development of Hans Moravec’s cart. The MIT AI Lab wanted their own robot so in 1973 he designed the “MIT Arm” which he commercialized as the PUMA robot (Vicarm, Unimation, Westinghouse, Staubli). The first, delivered to General Motors, is now in the Smithsonian collection. Fortune called him the “Father of the Modern Robot” (1980). More recently (2009), the IET (UK) named him “Godfather of Robotics”….hm-m-m.

Dan Swinehart for contributions to the SAIL programming language
The SAIL programming language and system was developed in the late 1960's by Bob Sproull and Dan Swinehart, with later contributions by Jim Low, Hanan Samet, Russ Taylor, Kurt van Lehn and others too numerous to mention. Derived from a class project, called Gogol, the language began with something resembling Algol-60 and then layered on contributions from many emerging language trends, including associative processing (based on Feldman's LEAP), records (typed compound data structures), references (typed pointers to same), support for multiple threads, and variable-length strings with automatic storage management, the latter inspired by a Bill McKeeman PL/I-like project on campus. The system was coded entirely in Phil Petit's FAIL assembly language to support hard-core systems applications for which LISP and other available languages were arguably inappropriate. Among others, notable well-known applications developed

in SAIL include Larry Tesler's PUB and early versions of Don Knuth's even more ambitious TEX document composition systems.

Larry Tesler for creating the PUB document compiler
In 1971, Les Earnest recruited Larry Tesler to create a document compiler that would go well beyond RUNOFF by supporting advanced publishing features. The software Larry built during the ensuing six months featured automatic numbering, headings, multiple columns, figures, footnotes, front and back matter generation, and cross–references. Its power was unprecedented. It also was evidently the first document compiler that provided for embedded spreadsheets. Today, we would call it a "scriptable markup language". The scripting language was a subset of SAIL. In that pre-SGML era, the markup syntax was non-uniform but it did allow arbitrary text to be bracketed by tags.

Because it was written in SAIL and because its syntax required use of the entire SAIL character set, the audience for PUB was limited. Nevertheless, at ARPANET-connected universities with PDP-10s, many a thesis was formatted using PUB. Because the code was open-source, Russ Taylor added FR-80 microfilm output and Rich Johnsson of Carnegie Mellon University (CMU) added font capabilities.

As with other markup languages, the output was often difficult to predict. At least two PUB users reacted to these shortcomings by developing better languages. Brian Reid, then at CMU, developed Scribe for nontechnical users. He implemented the first version entirely in PUB. Don Knuth developed TeX for authors of mathematical texts. Meanwhile, SGML and C took over as the basis for most future markup and scripting languages, and PUB became a forgotten milestone in digital publishing history.

Martin Frost for creating the first network news service
Martin Frost, with input from John McCarthy and Les Earnest, created two successive news services that each was the first of its kind. Beginning in 1972, APE could be used either to connect to the Associated Press newswire or to search recent stories based on combinations of pre-selected keywords. Beginning in 1974 NS (for News Service) indexed and stored stories from both the Associated Press and New York Times newswires and allowed users to either search for recent stories using any combination or words or leave a standing request to be notified whenever a story appears that contains the specified words. NS was widely used by people on ARPAnet for general news information.

During the Three Mile Island nuclear incident in 1979 the emergency response team at Lawrence Livermore Lab found that they needed up-to-the-minute information on developments at the site but couldn’t get it until they were provided with access to NS.

During the Tiananmen Square protests in 1989, Chinese students in the U.S. wanted to pass information to friends in China but there were no Internet connections there then. NS was then set up to locate news about China and forward it to a student distribution list so that they could print the stories and fax them home.

Commercial news services were developed later that now provide similar functionality to anyone who wants it and is willing to pay a subscription fee.

Phil Petit for initiating the first interactive electronic design system, SUDS
Phil Petit together with David Poole and Dick Helliwell initiated the development of the Stanford University Drawing System (SUDS) in 1969 as part of their Super Foonly

computer design project. It was the first interactive computer aided design system used for the design of a real computer and allowed designers to do both logic drawings and physical layouts on printed circuit cards and cross-check them for consistency. When the design was complete it produced artwork for printed circuit cards and backpanel wiring instructions that would control an automatic wiring machine. Dick Helliwell subsequently took over further development and maintenance of SUDS and went with it to the Digital Equipment Corporation, where it was used as their primary design tool into the 1990s. It was also used by Information International Incorporated (III), Daisy Systems Corporation, Foonly Inc., Valid Logic, and Cisco Systems. SAIL user Andy Bechtolsheim used it to design both the original SUN (Stanford University Network) workstations and all of those manufactured by Sun Microsystems through the mid-1990s.

Steve Russell for creating SPACEWAR, the first videogame
There were several early board games that ran on digital computers, such as Tic Tack Toe, and "Tennis for Two" ran on an analog computer. However the first dynamic videogame was Spacewar!, as reviewed at http://www.computer.org/cms/Computer.org/ComputingNow/computingthen/2009/03/CT-Lowood.pdf. It was originally developed for the DEC PDP-1 computer at MIT by Steve Russell and his colleagues in the Tech Model Railroad Club. Spacewar! spread through PDP-1 installations, and many others who had different computer and a display wrote versions for the equipment at hand. Steve then brought it to Stanford when he moved here to join John McCarthy and he and others then improved it. Meanwhile a company called Atari was formed to convert Spacewar into a commercial videogame called "Computer Space" using TTL electronics and no programmable computer. However Bill Pitts of SAIL beat them to it using a PDP-11 computer to create "Galaxy Game" and put it into the Stanford coffee shop and a local bowling alley. While the Galaxy Game was quite popular, Atari observed that their version of Spacewar was expensive to reproduce and somewhat hard for people to learn, so they instead introduced the game of Pong which was cheap to make, easy to understand, and a great commercial success. This allowed Atari to thrive for a time.

Lynn Quam for creating image comparison techniques for detecting time variable surface features on Mars.
Lynn Quam and his colleagues developed an image rectification and differencing techniques to facilitate identification of surface features that changed over time. This was used by Carl Sagan and other astronomers who visited SAIL every few weeks to evaluate satellite photographs of Mars. Quam successfully solved the problem of detecting small changes in the planet surface in the presence of changes of viewing geometry and illumination. His system was subsequently applied to pictures of Mars taken by the Mariner 9 orbiting spacecraft while the mission was in progress.