Inside SRI

Richard Waldinger's coffee-and-cookies, served in his office at 4 each day, is a regular feature at SRI's Artificial Intelligence Center. "It started in 1970 when I'd make coffee for a few friends," says Richard, a member of the center, "but over the years more and more people dropped in." Colleagues and visitors gathered here are: Front, on floor, Todd Davies (left). Fernando Pereira, (behind him) Mike Georgeff; second row, from left, Enrique Ruspini, David Israel, Martha Pollack, Doug Appelt, Paula Nokes (visitor), Richard Waldinger, Sandy Pentland, Amy Lansky, Oscar Firschein; back row, Paul Martin, Lorna Shinkle, Jean Pierre Muller (visitor, glasses), Jerry Hobbs, Yvan LeClerc, Leslie Kaelbling, and John Bear.

Published by
SRI International
Volume 1
Number 3
November 1986

95-75-659

From ERMA to CANDIDE:

(This is the first part of a two-part article on the Computer and Information Sciences Division.)

Whether it is someone like Jack Goldberg, who has been a computer scientist at SRI for 30 years, or Fernando Pereira, who has been here for just over 3 years, the people in SRI's Computer and Information Sciences Division are drawn to it as much by its achievements as by its atmosphere of inquiry and invention. The division has had an impressive number of the computer world's pathfinders and trailblazers, comprised variously of computer theorists, inventors, university professors, and entrepreneurs.

Goldberg directed the division's Computer Science Laboratory for over two decades (1960-83), following his contribution to the logic design of ERMA. ERMA was the pioneering bank-automation computer system SRI created in the late 50s for Bank of America. It, along with MICR (Magnetic Ink Character Recognition), revolutionized check handling and bank record keeping.

Jack Goldberg

Pereira is a native of Portugal and a Ph.D. from Edinburgh University. For him, the opportunity to rub shoulders and swap ideas with the array of talents collected in his division is maybe the richest reward for staying at SRI. Pereira's current projects in the Artificial Intelligence (AI) Center include the design of a natural-language database interface called CANDIDE. He says: "It would be hard for me to exchange this place for anywhere else."

The breadth of the division's research and development in computers and communications is very hard to match in other peer institutions. It ranges over the whole spectrum from pure to practical, from work that engages such people as philosophers, linguists, mathematicians, and physicists, to work that addresses practical problems like helping the military revamp its entire command, control, and communications computer system.

The division, headed by Don Nielson, is the largest in SRI, with a budget of $33 million this year. Its 260 members are organized into six centers:

- **Computer Science Laboratory** (headed by John Rushby)
- **Artificial Intelligence Center** (Stan Rosenschein)
- **SRI/Cambridge Computer Science Research Center** (Robert Moore)
- **Information Sciences and Technology Center (ISTC)** (Mike Frankel)
- **Network Information Services Center (NISC)** ("Jake" Feinler)
- **Special Communications Systems Laboratory** (Niles Walker)

Each of these diverse centers has its own strengths, whether in rate of growth and size of revenues, or number of papers published, or professional honors and affiliations.

The division belongs to a rather special set of defense contractors by virtue of the fact that over the last 20 years its people have been awarded perhaps $100 million by DARPA alone for work in computer and information science and technology.

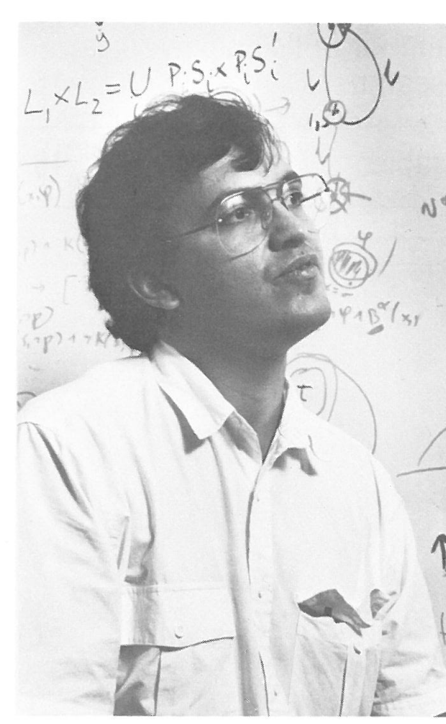
Some prestigious jobs in the computer world are filled by former division members. One former colleague, Nils Nilsson, is now chairman of Stanford's Computer Science Department. He was director of the division's Artificial Intelligence Center until 1984, and had been at SRI for some 20 years. This past August, another colleague, Barbara Grosz, left to teach artificial intelligence as Gordon McKay Professor of Applied Science at Harvard. She had been with SRI since 1973, and co-founded, in 1984, Stanford University's Center for the Study of Language and Information (CSLI), an enterprise that brings together computer scientists, linguists, mathematicians, and philosophers.

Quite a few professors leave tenured positions to come and join the SRI division. Among them:

- Joseph Goguen, former professor at UCLA. At SRI, he can devote more time to extending ultra-high-level programming languages, such as OBJ, which he created. He is also designing and building a new type of computer that will run the new family of "logical programming languages" he is developing. Goguen's project is an international one, involving researchers from centers in England, France, Italy, and Japan. He also heads the Semantics of Computation Group at the CSLI.
- Dorothy Denning, former professor at Purdue. She leads an SRI project on database security. She is author of a leading book on cryptography (the science of encoding secret material), and is president of the International Association for Cryptologic Research.
- Ray Perrault, former professor at Toronto University. He now heads the Natural Languages Program of SRI's Artificial Intelligence Center, and is president of the Association for Computational Linguistics.
- Franklin Kuo, former professor at Hawaii University. He headed the project that developed ALOHANET, the pioneering packet broadcasting radio network that was a precursor of two major Department of Defense research programs, Packet Radio and Packet Satellite. In 1976-77 he was Director of Information Systems in the Office of the Secretary of Defense. He is now associate division director as well as executive director for Asian Programs for the Engineering Research Group.

"Why give up a tenured position for SRI?" In the case of Kuo, it was the fresh challenge. He was a full professor at the age of 32, and 16 years, 6 books, and many papers later he was still a full professor. "I was tired of teaching. What I like about SRI is that it combines the atmosphere of a large industrial research lab with the intellectual freedom of a university."

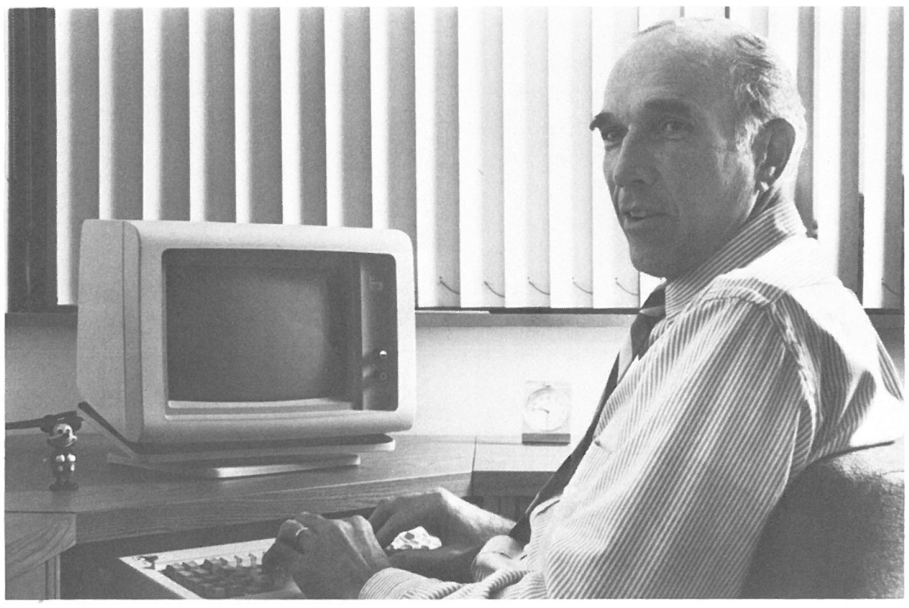
A lively interchange continues



Fernando Pereira

The Computer and Information Sciences Division spans the pure and the practical, the present and the future

between the university and SRI. Kuo is consulting professor of electrical engineering at Stanford. Barbara Grosz was consulting associate professor in computer science and served on the dissertation committees of eight Stanford Ph.D. students. Other consulting faculty include Stan Rosenschein, Ray Perrault, and Fernando Pereira, who each have taught a course at the university. On the other side, Stanford students often fulfill some of their course requirements by working at SRI. SRI is one of three partners in the Center for the Study of Language and Information. The SRI-university interchange is active on several levels.



Don Nielson

Knowledge as product

"We like to have a significant part of our work be basic research," says Don Nielson, the director of the Computer and Information Sciences Division. "Much of what we produce is knowledge but we also deal in more tangible products."

Of the division's six groups, three are, historically, the "computer" side of the division (Computer Science Lab, AI Center, Cambridge Center). They are oriented primarily to basic research. The other three groups are historically the "communications" or "information" side of the division, more oriented towards applications. ISTC does both basic and applied research. NISC is more service oriented, with some R&D efforts. Special Communications conducts special development projects for the U.S. government.

"While the division delves into a wide range of computer research topics, one specific thrust is to bring about the marriage of computers with communications," says Nielson. He has personally played a leading role in joining the two at SRI. He headed SRI's Telecommunications Sciences Center when it was merged in l978 with the Computer and Artificial Intelligence groups to form one division. This division was enlarged in 1985 with the addition of AITAC (Advanced Information Technology Applications Center, under Mike Frankel), and further enlarged this September with the addition of the Special Communications Systems Laboratory. The recent formation of the Cambridge Center results in the present division. The division is funded predominantly by the U.S. government to fulfill two missions:

l) Conduct basic research in computer and information sciences, and

2) Develop advanced applications of the newest research results.

Ultimately the reputation and strength of the division comes from its project work over the past 25 years. On the computer side, some of the talents that created ERMA, for example, went on to create a fault-tolerant computer, called SIFT, and the HDM software design methodology, and that work laid the foundation for our present strength in computer architecture, software methodology, and database security and safety. On the communications side, much of our past work on ARPANET and packet radio networks and more recent efforts like ADDS and ADDCOMPE have paved the way for a new $37 million project for the Army called High Technology Research and Development. Of course, the two streams of activity sometimes converge, blurring the line dividing "computer" from "communications" or "information."

In this issue of Inside SRI we will look at the computer side. In the next issue, we will continue with an account of the communications side.

Frank Kuo

Computer science

Concern with methodology

Paul Martin, a computer scientist in the division, has a reversible sign hanging outside his door. At times, the sign announces: "Mr. Hyde is fixing bugs. Please come in." At other times the sign is turned around and says: "Dr. Jekyll is doing science. Mr. Hyde will be back 11:30 to 4."

Doing Science and Fixing Bugs wryly encapsulates two of the uppermost concerns of computer science at SRI ever since its beginnings.

Computer science, artificial intelligence, and "computing augmentation of personal work" were three SRI themes of the early 60's which developed into major programs of the present division. The first two continue under divisional laboratories while in the 70s "computing augmentation" in part joined Tymshare Corp. and in part merged with the Telecommunications Center. We shall look at each theme in turn.

The origins of the computer science laboratory goes back to ERMA in the early 50s. Heading the team was Jerre Noe, now professor at the University of Washington. Many people worked on the ERMA project, among them three who did the primary logic design for ERMA, William Kautz, Bonnar Cox, and Jack Goldberg.

"There was no science then, it was all technology," recalls Goldberg. "After the project was over we in computer design thought the world needed a more scientific method of designing computers, and so Bill Kautz established the computer science program to do research in mathematical methods for computer design."

This was 1957. The program initially was part of the Computer Techniques Laboratory.

Hardware

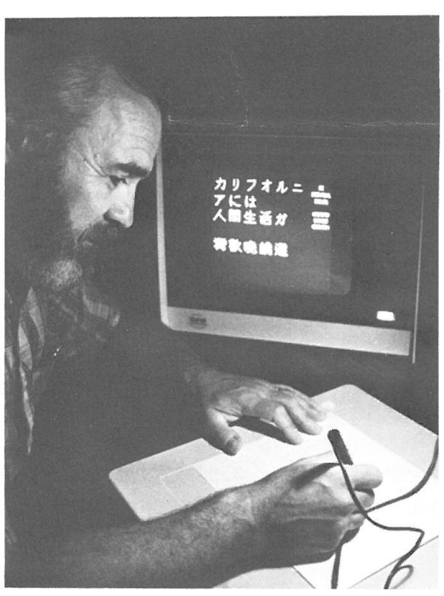
For about 10 years the directions of the program emphasized computer hardware: how to build machines and how to make them reliable. It was not until around 1970 that the focus began to shift to software.

Continued next page

Computer science,

continued

The computer would not simply be an advisor, but a very dependable controller.



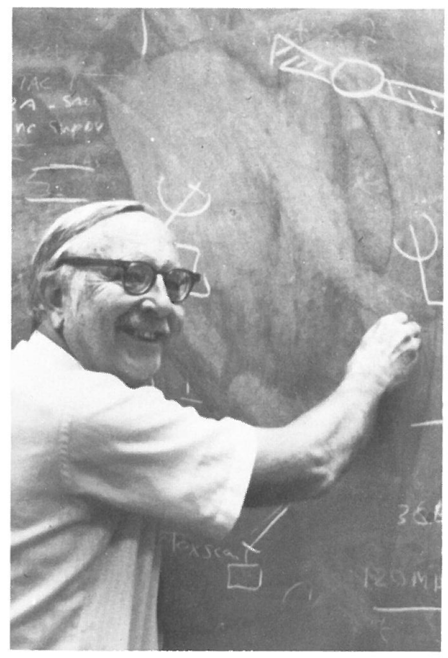
Hewitt Crane

The program's first research effort was in switching theory. There was growing interest in using computers for such applications as telephone switching, train control, and control of spacecraft, but the early computer elements would fail too frequently, and the new semiconductor technology was thought to be too unreliable. So one of SRI's first large research efforts was in an alternate technology, All-Magnetic Logic. Hewitt Crane, a member of the Computer Techniques Laboratory, had a notion, based on work he did at RCA Research Labs., for an extremely reliable computer made not of semiconductors but of magnetic elements. Crane called his concept MAD (Multi Aperture Device). MAD was used in small computer circuits and special purpose computers, but the technology finally lost out to the transistor. Crane, however, continues to be one of the most inventive scientists at SRI, having invented such successful devices as the eyetracker and the optical pen. Crane is now director of SRI's Sensory Sciences Research Lab. Goldberg calls Crane, Charles Rosen (founder of the AI Center), and Doug Engelbart (founder of the Augmentation Research Center) three giants of the early days of SRI's computer work. (We will return to Engelbart in the next issue of Inside SRI).

Meanwhile the program also did research in the theory of "fault tolerant computing," which was a blend of computer architecture and computer logic. It started with a study of reliable logic networks sponsored by the Jet Propulsion Laboratory, and went on to do a series of studies for NASA on techniques for designing ultra-reliable spaceborne computers.

Cellular logic

The program soon saw that integrated circuits were moving towards decreasing size and increasing complexity. There was talk of getting a computer on a chip. One instance of work that anticipated the future was the development of a program based on "cellular logic," led by Robert Minnick, William Kautz, and Bernard Elspas (Minnick is now a program manager at National Science Foundation). The idea was that all the transistors on a computer chip would be organized in a very regular way, like a checkerboard. Some of the concepts in the pro-



Bernard Elspas

gram became popular some 10 years later as "systolic logic", wherein the activity of computer elements would flow down a channel in pulsating waves, like that in a living organism.

By 1970 the Computer Science Laboratory saw the problems of computer software growing rapidly. The problems stemmed from the inherent weaknesses of the available computer languages.

"Popular programming languages like FORTRAN and COBOL are at their core fairly easy to understand, but if you use them to build practical programs you have to write many many lines to get your job done," explains Goldberg. "Everytime you write another line you introduce new possibilities for mistakes and you also make the thing much more difficult for another person to understand.

"Before you use the program there is an enormous effort that has to go into fixing mistakes. Debugging programs is extremely expensive and even after you do it there is always a residue of mistakes that will continue to haunt you."

"This problem has reached a crisis over the last 15 years," Goldberg continues, "because new ways of using computers require very complicated programs to be written. Unfortunately, the languages are so terribly primitive that there's a great conceptual gap between our wishes and our ability to realize those wishes in a program. The cost is high, failures are frequent, and computers can't be trusted—a real crisis."

Proof of correctness

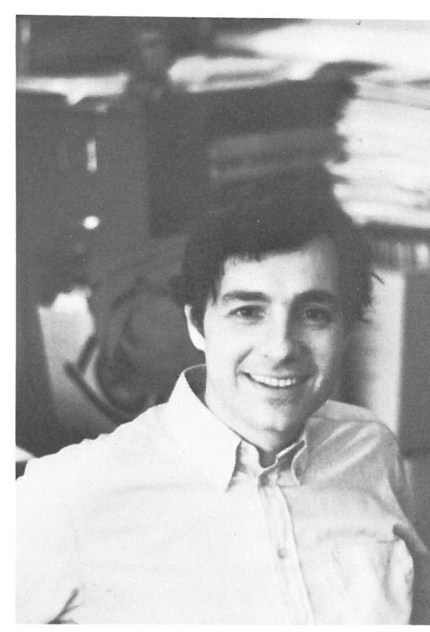
One of the first things the laboratory did to respond to the software problem was to apply mathematical methods to "proof of program correctness," to prove what a program will do for all its allowed inputs without having to run it. This was a large effort that started under the leadership of Bernard Elspas, Karl Levitt, and Abraham Waksman. Some major work was done by Robert Boyer and J Moore (both now at the University of Texas-Austin) and Robert Shostak and Richard Schwartz (later co-founders of a software company, ANSA).

Another response to the software problem was the notion of software methodology. A young man named Larry Robinson joined the SRI laboratory in 1973 and used the ideas of his professor, David Parnas at Carnegie-Mellon University to organize large software systems in modules, whose functions and inter-relationships could be easily analyzed. Robinson called his work HDM (Hierarchical Development Methodology).

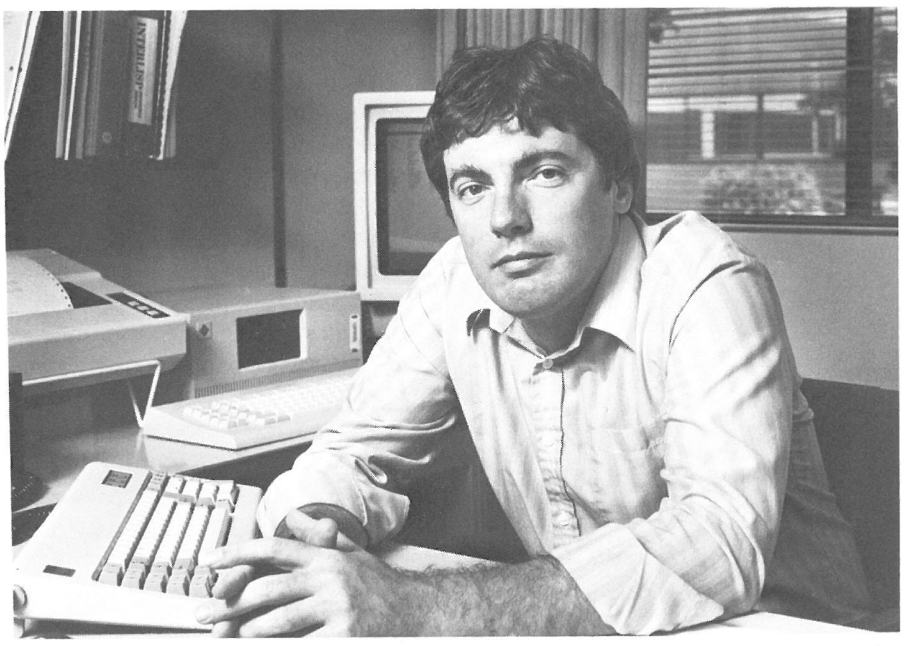
HDM and the-proof-of-correctness work contributed to the group's computer security work. They were basic ideas for PSOS (Provably Secure Operating System), which came to be recognized as a pioneering idea in operating systems. The PSOS effort (1974) was led by Peter Neumann. It brought together Neumann's experience in operating systems, Robinson's in program methodology, and Karl Levitt's in program proving.

Designing and verifying secure systems has since become a very important field and a major area of activity in SRI's computer science group today. Also about 1974 NASA

Continued next page



Karl Levitt



John Rushby

The basic and the relevant: Mathematical methods

Today computer reliability and the related concerns of fault tolerant computing, computer security, and program methodologies are still the basic themes of the Computer Science Laboratory.

"Our rather small staff of 20 researchers, mostly Ph.D.s, maintain two or three traditional areas of excellence," says Director John Rushby. "We do unclassified pure research. However, relevance to real-world problems is important to us, both from a moral and a financial viewpoint, and we have in many cases slanted our work to the concerns of a particular DOD client." Software is the main emphasis because software is that which above all controls the computer. Commenting on the strong mathematical orientation, Rushby says: "If you are going to trust your life or economy or national security to computer programs, you ought to make sure they are constructed in a way that gives you some conviction that they work right. Current methods of construction don't enable you to have that kind of conviction."

"The way forward is to do what other engineering disciplines do, that is, develop an appropriate mathematics so that you can analyze your programs. An engineer doesn't find out whether planes can fly just by flying them and see if they fall out of the sky. He does such things as mathematical modeling and stress analysis of the wings and other components, and uses such mathematical methods as finite element analysis."

The engineering mathematics of computer programs is mathematical logic, Rushby says, and the main thrust of the lab is to develop that mathematics and its application in computer programs.

Real problems: Snoopers and Trojan Horses

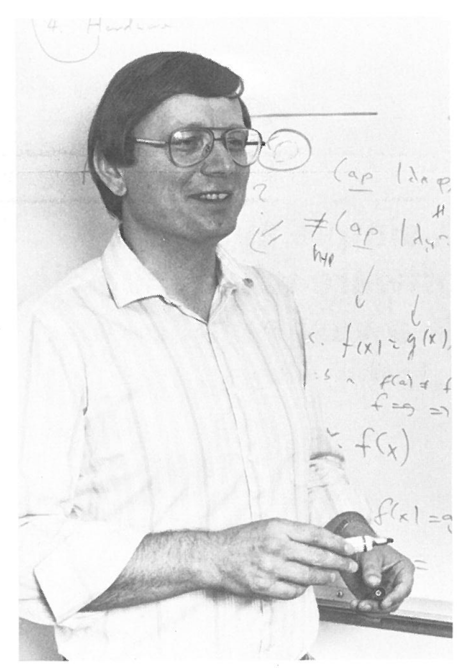
One large current project, E-HDM (Enhanced Hierarchical Development Methodology), is in the area of "formal specification and verification": You construct a precise mathematical ("formal") specification of the properties you require of a software system, then demonstrate by rigorous mathematical techniques that the system has those properties, rather than rely on testing and other more fallible methods.

Before formal methods were introduced software was developed using "informal" methods, that is, specifications were stated in plain English. "Formal" language, or the language of mathematics or logic, is more precise and more machine-processable. One can use the machine to reason about formal specifications and analyze the program for syntactic and semantic consistency. One can, in other words, generate theorems from programs and then prove those theorems with a mechanical theorem-prover.

The E-HDM project is headed by Friedrich von Henke and funded by NSA (National Security Agency). Von Henke's group has developed a language, E-Special (Enhanced Special), for formal specification and verification.

Because of client interest, one of the properties receiving the greatest attention is security. Is a particular program or parts of the program secure from unauthorized persons—snoopers? E-HDM is developing the method to prove that the program

Continued next page



Friedrich Von Henke

Continued from page 4

made a challenge: put together all of SRI's ideas and theories on fault tolerant computing into practice, that is, build an experimental computer that would be extremely fault-tolerant. NASA wanted a system that could be used to control safety-critical functions of airplanes. The computer would not simply be an advisor, but a very dependable controller.

An enormous success

This became the SIFT program, which lasted 10 years. SIFT (Software Implemented Fault Tolerance) was conceived by John Wensley (who later founded August Systems Corp.). SIFT was an architectural scheme that depended heavily on software programs. There were other designs that did similar things in fault tolerance, but the mechanism for doing it was usually hardware.

SIFT was an enormous success. First, the design was effective and simple, so that its principles could be realized in many different forms. Second, it demonstrated a new idea in computer design, that one could use mathematics to prove a design to be correct. Third, the group uncovered a number of research problems in the design of "distributed" systems that opened up a whole new line of research. One of these is the so-called "Byzantine Generals" problem, first described and proposed by SRI's Leslie Lamport. In Lamport's model, computing in a network of unreliable computers is made analogous to the situation of a group of generals in a Byzantine army who fear that one of them is a traitor. The Byzantine Agreement problem has taken on a life of its own and is now one of the hottest topics of research in distributed computing. □

has security, among other properties.

"We took our names, E-HDM and E-Special, from the older system HDM and its language Special, but we are otherwise unrelated," says von Henke, who obtained his Ph.D. in applied mathematics from the University of Bonn. "Our approach is radically new and different from HDM and other approaches elsewhere."

E-HDM is a prototype system which will serve as a tool for software engineers, those programmers who design programs to the requirements spelled out by clients. The client may be a government agency, a chemical or nuclear plant, a business corporation, or any other kind of user.

Dorothy Denning, who heads another large project concerned with computer security, is developing a method for designing databases to foil not only snoopers but also "Trojan Horses".

"The reason why people can break into a system is because it has weak spots they can penetrate," says Denning. "It's because the software isn't designed right from the bottom up."

"A simple system of software will generate a vast number of putative theorems..."

Denning's group is developing a concept called "Secure Dataviews" for handling classified information in databases. The database system would meet the most stringent criteria for security, called "A-I", laid down by the National Computer Security Center.

The method starts with a mathematical model of the database system that defines the behavior of the system, especially with respect to all properties relevant to security. Then design specifications would be derived, and the design would be verified for consistency with the specs.

The approach is called "Secure Dataviews" because users do not see the whole database but only particular views of the database. Instead of putting security controls on the physical data, the controls are put on the views. For example, a given employee of a company may have a view of the database that includes names and addresses of other employees but not their salaries.

Another large aspect of the "Dataviews" research is data integrity. The database must be verified

Dorothy Denning

for correctness and consistency: making sure that certain relationships among the data are satisfied. For example, there may be a rule that says that salaries related to a certain piece of data is always between $20,000 and $100,000. Another important aspect is database reliability—that if the system fails one can bring it up again.

The dataviews method will not be classified. Then won't it be easy for a user or Trojan Horse to penetrate the system? No, says Denning. "There's always an assumption the enemy knows your method for protecting the data. By taking a formal approach, we can be sure the mechanisms cannot be circumvented."

Other avenues: Through pictures

"The trouble with formal specification and verification is that it is staggeringly expensive," says Rushby. "It costs millions of dollars to prove a fairly small system. A simple system of software will generate a vast number of putative theorems which must be shown to be true in order for the system to have the properties you've asserted it has. It's very expensive in terms of the skilled manpower you need for the sheer number and intricacy of these theorems, which are astonishingly difficult and subtle to prove. One needs a very talented person, usually a Ph.D., who knows all the math, logic, and system design, to drive the theorems through a mechanical theorem prover."

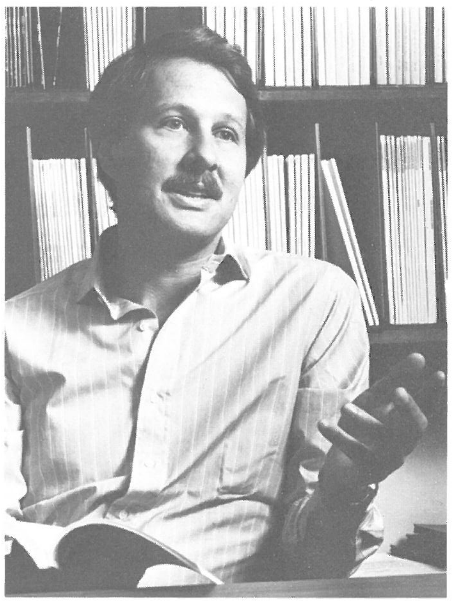
So specialized approaches to formal specification and verification are being explored. These approaches are aimed at checking only limited properties of a computer program. An example of such work is a project called "PegaSys."

PegaSys, a software package created by Mark Moriconi, has received wide attention, and a paper Moriconi wrote on it won a Best Paper Award at the I985 SIGPLAN Conference of the Association for Computing Machinery.

PegaSys supports a formal specification language that is easy to read and write and reason about mechanically. The language is pictorial and is used to document how the pieces of a large program are assembled. A user constructs a specification through structured graphics operations. A program is described by a hierarchy of pictures. These pictures have mathematical meaning, enabling PegaSys to understand them and to prove simple theorems about them, including that the pictures used are an accurate representation of the structure of a program.

PegaSys's special importance is in software maintenance, since 80 to 90 percent of the cost of software is in maintenance, not construction ("maintenance" applies to any modification of the software.)

"Someone who maintains a program has to know how the pieces are put together, what depends on what, and this is usually never documented," Moriconi says.



Mark Moriconi

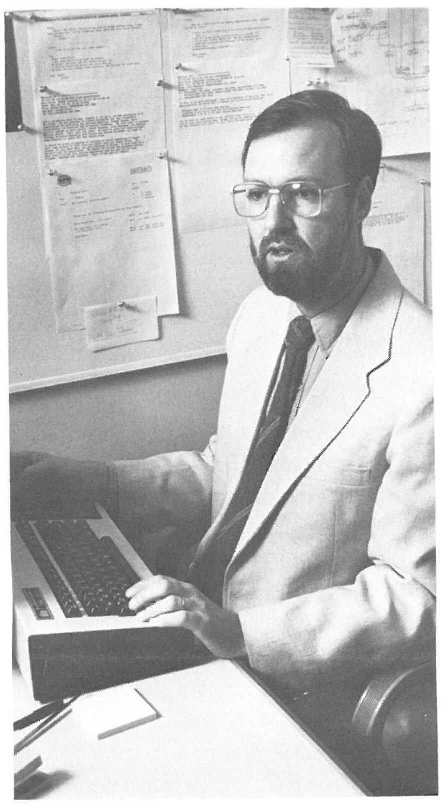
"PegaSys is easy to understand because the pictures are much more intuitive than the equivalent logical expressions. PegaSys can be used for program design too, but maintenance has been the big problem and most neglected in the past."

Through new languages

Ultra-high-level programming languages are languages that can express programs in a manner that is much closer to specifications than ordinary programs written in high-level languages like FORTRAN or

COBOL, which require the programmer to spell out every detail, which means, of course, more chances for mistakes. In ultra-high-level languages, the programmer does not have to spell out all the details; they are built into the language.

SRI's Joseph Goguen has developed over the last ten years an ultra-high-level language, OBJ, based on equational logic and the use of the term-rewriting system to mechanize the logic. ("Term-rewriting" is jargon from logic for the substitution of equals for equals.) Goguen is now heading a team that is developing a family of "logical programming languages" that will bring together the best features of three major new programming styles: logic programming (pure PROLOG is an example), object-oriented programming (e.g., SIMULA and SMALLTALK), and functional programming (e.g., pure LISP, FP, and OBJ).



Joseph Goguen

SRI has also been engaged in the development of another ultra-high-level language, LUCID, the creation of Ed Ashcroft of SRI and Bill Wadge of the University of Victoria, Canada. It is a functional programming language based on "data-flow" principles.

Ultra-high-level languages, though powerful and expressive, tend to execute slowly on traditional computers, which are based on "von Neumann architecture," where processing is sequential. The limitations of imperative languages like FORTRAN are due basically to the fact that they were designed for that architecture. For OBJ and other ultra-high-languages to work optimally, they need their own machine architectures. As Ashcroft, Wadge and others have pointed out, "The software problem is really a hardware problem."

Goguen's team is developing the architecture for a new kind of computer called a "Rewrite Rule Machine." It will exploit the parallel or concurrent computing made possible by the reduction in size and price of computer hardware. Hundreds and thousands of transistors can now be put on a single chip, and the processors that they constitute can be run concurrently.

"...lots of things can happen in parallel."

"There is good reason to believe that we will succeed in building a machine to run ultra-high-level programs at very high speed because these programs don't burden you or the computer with details about how things are to be done," Rushby says. "They leave a great deal of flexibility and opportunity to the computer, the evaluation mechanism, to do things in whatever order is convenient, and that means lots of things can happen in parallel." □

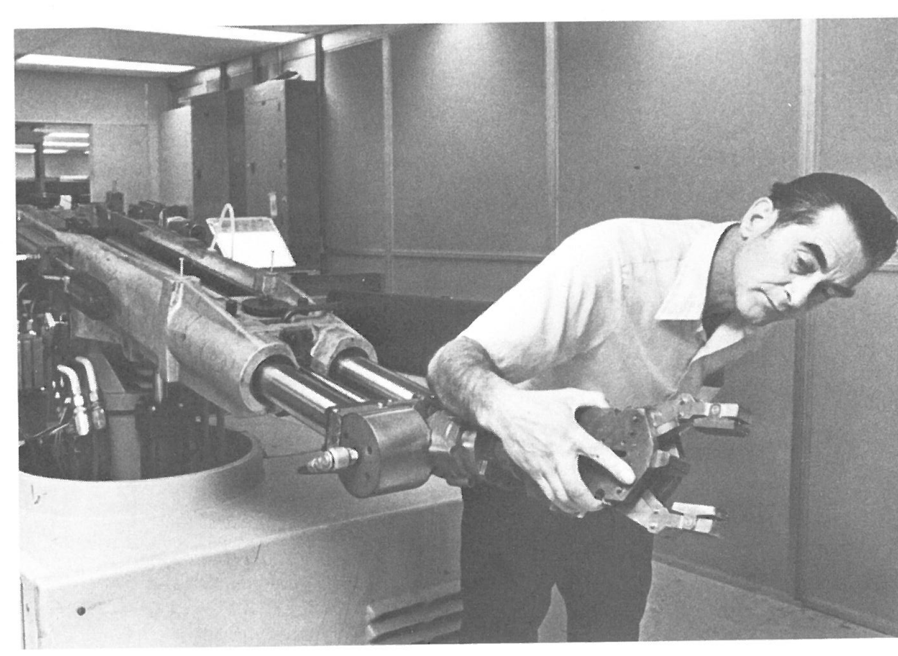
Artificial intelligence

Learning machines and reasoning robots

SRI's work in artificial intelligence over the last 25 years has made it one of the foremost AI centers in the world.

Charles Rosen and Nils Nilsson were two of the most influential figures of the first 20 years (1964-84) of SRI's AI work. Collaborators at first, their paths later diverged, with Rosen going towards robotic applications and industrial robots, and Nilsson going towards AI theory, especially logic and reasoning. Rosen left SRI in 1978 to found Machine Intelligence Corp.(MIC), a maker of vision modules and robot systems. He was perhaps the first to propose, as early as 1963, a robot that could "see" and "learn" and move around with the help of TV cameras and sensors. The Industrial Automation Program he founded at SRI played a leading role in helping industry enter the robotics age. Nilsson, now head of Stanford's Computer Science Department, is generally recognized as one of the world's leading figures in AI.

Many others made outstanding contributions to the AI program. Some of them have moved on to other positions in the computer field, where their influence is wide and far-reaching (see Spin-offs, next page).



Charles Rosen

Charles Rosen is essentially the founder of the AI Center. His interests also led to the present SRI programs in Electron Physics and Industrial Robotics. A physicist who joined SRI's Applied Physics Laboratories in 1957, he had two main interests, micron-size electrical devices and "organizing systems" ("intelligent circuits"), a concept that foreshadowed computer-based AI. These interests led him to build one of the first learning machines, MINOS I; it was based on the "perceptron", an idea of Frank Rosenblatt, a Cornell psychologist. The perceptron was a model of neural circuits; variable electrical resistances simulated variable synaptic strengths between neurons. MINOS I could be trained to recognize patterns, that is, it could "learn."

In 1963 Rosen conceived and proposed the idea of an "automaton" (robot): a learning machine equipped with television cameras ("eyes") and other sensors, and with motorized effectors ("arms and legs"). As interest in computers grew, the AI group also decided to add computer programs to the machine, which eventually became Shakey the robot, completed in 1968. Later, Rosen became interested in applying pattern recognition to industrial parts-assembly and inspection, and started an industrial robotics program.

Among the pioneering achievements of the first 20 years are: MINOS I-III, 1961-68 (a learning machine); QA 1-4, 1969-70 (a question-answering system based on machine reasoning); SHAKEY, 1966-72 (a mobile robot equipped with programs for vision, planning, and learning); PROSPECTOR, 1977 (an expert sytem for prospecting mineral deposits); and LIFER, LADDER, TEAM, 1976-81 (software tools for natural language dialogue with computers).

Spin-offs

Artificial Intelligence Center

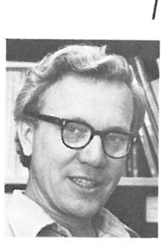
Charles Rosen
Chief Scientist & Founder, Machine Intelligence Corp.
(vision modules & robot systems)

Gary Hendrix
Founder & VP, Semantec
(Nat.lang.software for microsystems)

Barbara Grosz
Professor, Appli. Sci. Div.
Harvard University

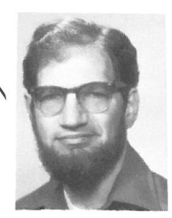
Cordell Green
Founder and Director, Kestrel Inst.
(non-profit; use of AI in software engineering)

Richard Fikes
VP—Research
Intellicorp
(AI software tools)



Nils Nilsson
Chairman, Comp. Sci. Dept.
Stanford University

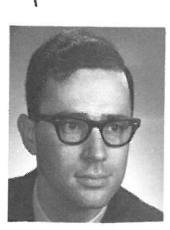
J. Martin Tenenbaum
Director—AI
AI Laboratory
Schlumberger



Bertram Raphael
Director—R&D
Info. Resources
Hewlett-Packard

Earl Sacerdoti
VP—Chief Exec. Office
Teknowledge
(expert systems & applied AI)

Daniel Sagalowicz
VP—Prod. Development
Teknowledge
(expert systems & applied AI



Peter Hart
Co-founder & VP
Syntelligence
(expert sys. software)

Richard Duda
Co-founder, Syntelligence
(expert system software)

Some former AI Center staffers and their present positions.

AI Center now

The Artificial Intelligence Center now has 50 researchers, almost all Ph.D.s. It is headed by Stan Rosenschein. Research is conducted under three programs, **Reasoning and Representation, Natural Language Understanding,** and **Perception**. The work ranges from the very theoretical to the somewhat applied.

"Although much of our work is aimed at specific applications, our real goal is advancing the state of the art," Rosenschein says. "We tend to measure our progress by the publication of papers in journals, conferences, seminars, and things of that sort."

Reasoning and representation

Bigots and robots

Winston Churchill has said: "Fanatics are people who can't change their mind and won't change the subject." Churchill would have been pleased to learn that this definition of a fanatic also nicely defines an unsophisticated robot.

Robots are getting more sophisticated, however. SRI's Artificial Intelligence Center and other peer centers are seeing to that. SRI was a trailblazer in robotry in the 60s with its robot Shakey. Now it is in the vanguard again with its new robot project, Flakey, son of Shakey.

"Flakey," says Center Director Stan Rosenschein, "will be a mobile robot that can exhibit both perception and reasoning, and do it in real time, something not achieved before. It will be able to interact with humans in natural language, and do complex tasks."

"Natural language" means English, not a programming language like PASCAL or LISP. A "complex task" in English would be something like: "Flakey, please take this document to Dave Wilkins."

Flakey will be clever enough to head for Dave Wilkins office to execute the task. However, should he meet Wilkins in the hallway, Flakey would be sophisticated enough to change his mind and give the document to Wilkins in the hallway.

At present robots can be extremely clever but not clever enough to change their mind in this fashion. They are characterized by singlemindedness. Specialization, if you will. Someone has said: "Specialization is the mark of a machine; it's humans that can be versatile."

"Right now robots are usually highly optimized along one dimension," says Rosenschein, who also heads the Flakey project. "Different robots have different talents." There

Stan Rosenschein

are now robot hands that can move so fast that the fingers can stir an egg. There are Japanese robots that can play the piano. Other robots are developed along the dimension of strength; still others along the dimension of mobility. Odetics, for example, has developed a robot that has legged locomotion and walks like an insect.

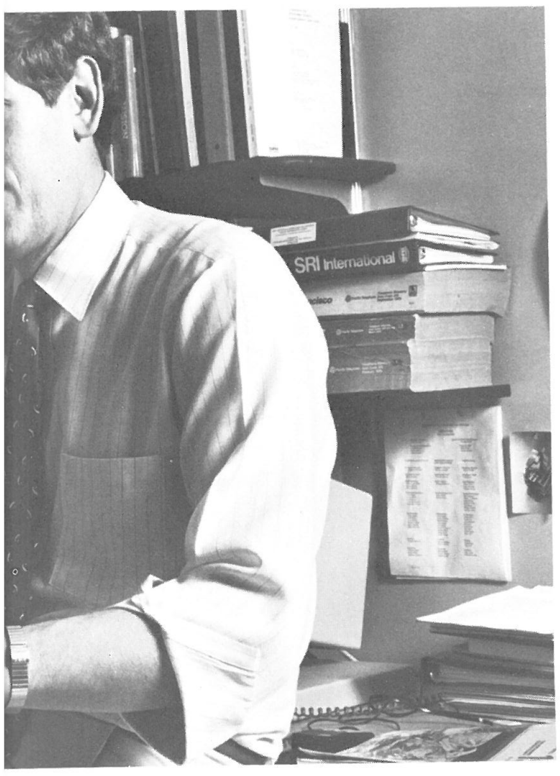
There are police robots that are sent out to explore burning buildings or radioactive sites like Chernobyl; or those sent to explore deep-sea objects like the sunken Titanic. However, all of these robots have a low degree of autonomy, says Rosenschein. The issue is autonomy, and Rosenschein's project is aiming at a much higher level of autonomy for Flakey.

Robots that go into fire and dangerous situations have to exhibit different responses in different situations. How do they decide what to do when? Because the technology doesn't yet exist to do anything else, one of the strategies has been to use "tele-operation": connect robot sensors to humans on the other end so humans can see what the robot sees and can give commands ("turn right, turn left") to the robot. "These robots are puppets more than autonomous agents," says Rosenschein.

"Get me some coffee"

Flakey is to serve as a testbed for bringing together "in one brain" much of the research that is being done at the AI Center. But Rosenschein is quick to point out: "We do much more than robotics at the Center, and Flakey is just one of many AI projects."

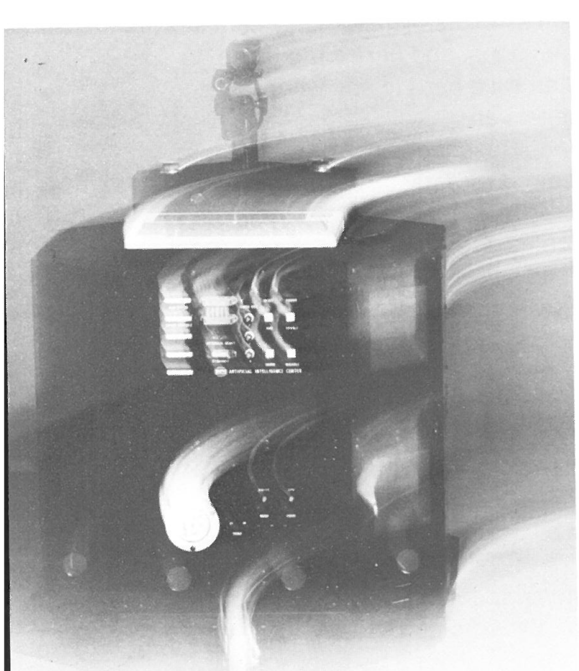
Flakey's home is in the Rea-



soning and Representation Program because the project's main focus is to get the robot to reason—understand the world and solve problems.

"One of the things our lab is noted for is its use of formal techniques such as logic to study machine reasoning," Rosenschein says.

There are different strands of AI, he explains. One important strand is the expert-systems community, involved in constructing large knowledge-base systems. Its programming paradigms were developed mainly under the influence of Stanford's Edward Feigenbaum, and they tend not to be primarily logic-based. Another strand, led by Nils Nilsson, John McCarthy, and others, emphasizes the use of formal logic for knowledge representation and reasoning.



Flakey

Flakey will be used for research in two areas of reasoning and representation: reasoning in an environment that changes, and reasoning in a multi-agent domain (that is, when several robots have to cooperate or when robots have to cooperate with humans.)

"Flakey, go and get me some coffee." What if, while he is getting it, a fire breaks out?

"You want it to be smart enough to forget about the coffee and put out the fire first," says Michael Georgeff, the director of the Reasoning and Representation Program. A former professor, Georgeff is Australian and has a Ph.D. from Imperial College, London. "You want it to have the flexibility of the human. But this is something AI has not seriously considered—what it's really like to have a machine survive in a real, dynamic world."

One project Georgeff's team is conducting is sponsored by NASA's space shuttle program. Currently when the space shuttle goes up it has malfunctions which have to be corrected by the astronauts. To do this they have to look up a complex series of instructions, contained in two or three fat books, thousands of pages thick. Meanwhile, two or three hundred controllers on the ground do exactly the same thing, double checking their work, and going over the same books. The space program would be greatly helped if relatively straightforward malfunctions, not requiring complex reasoning from first principles, could be handled by robots.

However, Georgeff believes that standard "expert systems" will not work.

A mind of its own

"The standard expert system is just not reactive enough," says Georgeff. "If you tell the system (robot) to fix the coffee pot, it will fix it before it notices a fire. It's too singleminded and static."

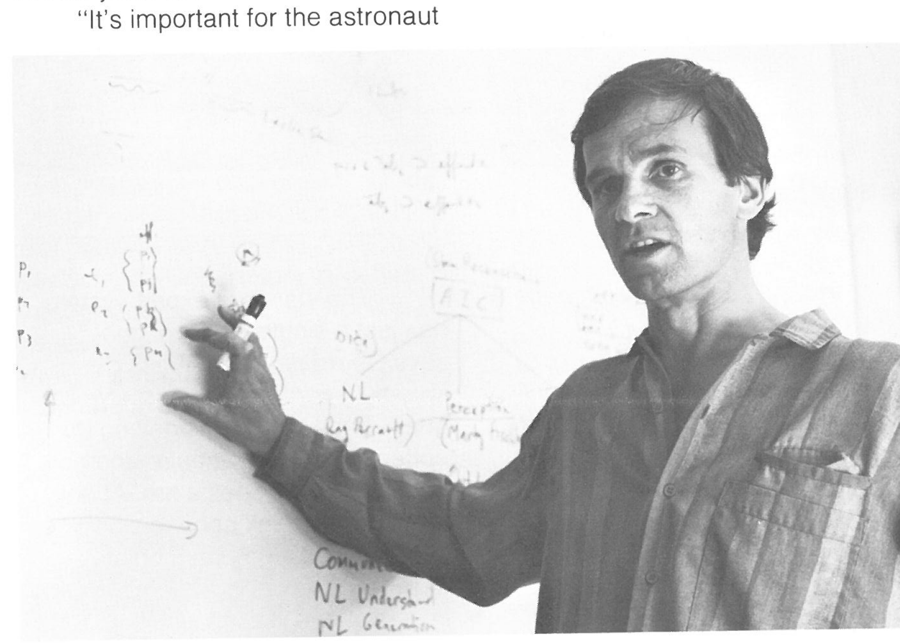
Also the current programming systems can't interact with humans in any sensible way. In the current programming languages, a conversation will go like this:

Astronaut: Please close the valve.
Robot: I can't close the valve.
Astro.: What happened?
Robot: I tried to close the valve.
(Or, I tried step 33).

That is all the robot can tell the astronaut. But what is needed, Georgeff says, is a system that is a "rational agent" or "astronaut's associate", one that can hold a dialogue with the astronaut like the following:

Astronaut: What happened?
Associate: Well, I was trying to close this valve, with the intention of isolating the fuel tank. I was doing this because I believe it's the best thing to do when there's a leak in the vernier jet.
Astronaut: I'm not sure your beliefs are correct. I believe you will find that if you look more carefully, the vernier jet is o.k.

"It's important for the astronaut to have this interaction with the system if it's to be truly useful," says Georgeff.

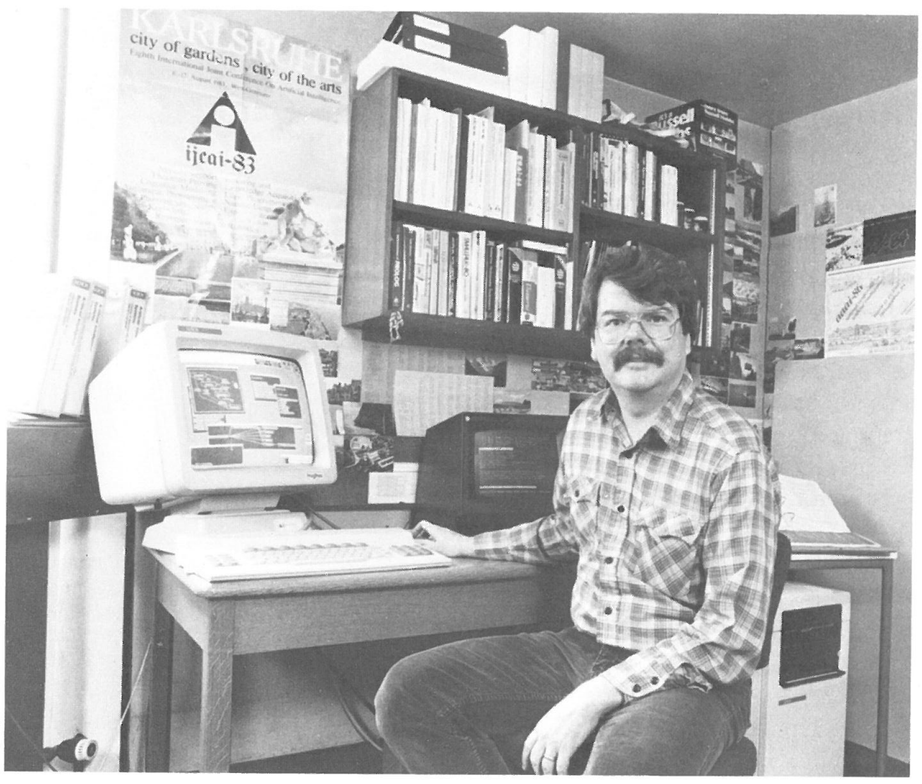


Michael Georgeff

In other words, the robot should have a "mind" or "cognitive architecture," which contains beliefs, goals, and intentions (B-G-I's). The reasoning system Georgeff's group is building is called a "procedural reasoning system" because the way the knowledge is represented to the computer-brain is in terms of procedures. It is a way of representing procedures that is more flexible than programming languages and more expressive than current expert systems.

Continued next page

The SRI/Cambridge AI center



Bob Moore

The SRI/Cambridge Computer Science Research Center, part of SRI's Computer and Information Sciences Division, was opened this year to carry out research into artificial intelligence, in collaboration with the University of Cambridge Computer Laboratory.

The Center was launched as a cooperative venture funded under the auspices of the Alvey Program, a government-backed program to foster research in the United Kingdom. Participants include SRI, the Alvey Directorate, and seven of the leading high technology companies in Europe: British Aerospace, British Telecom, Hewlett Packard, ICL, Olivetti, Phillips and Shell Research. Initial funding for the first program will be $1.8 million spread over three years.

The new center, directed by SRI staff scientist Robert Moore, will concentrate initially on the problems of processing natural languages like English in a form that computers can understand, and will focus on logic and formal methods, AI generally, and application programs and literature.

Commenting on the new project, David Shorter, director of the Intelligent Knowledge-Based Systems part of the Alvey Program, said: "I am particularly pleased that a consortium of companies in the U.K. has been set up to support work in natural language processing.

"SRI has probably the most experienced team in natural language research in the world at its Menlo Park headquarters. The link between this team and the new Cambridge center, coupled with inputs from other Alvey-supported university work in the U.K., should provide a rich environment in which the program of the consortium can really flourish." □

Continued from page 9

"The question is not how to do it but how to do it efficiently..."

"The standard expert system has just a single top level goal," says Georgeff. "In contrast, our 'rational agent' can have many goals. If the top goal is getting the coffee, it can also acquire another goal because it sees a fire. At any time it could think about a whole host of goals, and can mediate between many goals and conflicting beliefs. It can reason which goal to achieve first or how much time it should spend thinking about the problem. That's an issue the current commercial expert systems don't address."

Theorem-proving

Underneath all this work on reasoning and knowledge-representation is some basic research in theorem proving and program synthesis. Most of the robot's "knowledge"—beliefs, goals, intentions—is represented in a logic called "predicate calculus", and theorem proving involves developing efficient systems for using that logic to come to conclusions. "The question is not how to do it," says Georgeff, "but how to do it efficiently, a very complex problem."

Program synthesis is a kind of planning: one tells the system "what" one wants it to do, and the system figures out "how" to do it—it does automatic programming. SRI's Richard Waldinger has done much work in this area. "The trouble with program synthesis right now is, it takes as much time to specify WHAT you want it to do as to tell it HOW to do it," Georgeff says.

Also fundamental to this work is basic work in areas related to philosophy. Kurt Konolige is developing (for the Office of Naval Research) a formal reasoning system that can reason about beliefs both about itself and other agents. Some of the reasoning processes he has to deal with are captured in old philosophical conundrums like the Three Wise Men and The Unfaithful Wives.

Georgeff's team is also working with philosophers in Stanford's Center for the Study of Language and Information. Of interest are centuries-old questions on what it is to be rational. Is it possible to have contradictory intentions? Can someone have an intention to marry both Jane and Ann? How do you resolve conflicting intentions? How does an intelligent agent change its mind in a rational, way?

Where are the beliefs?

There are differences in approach between Rosenschein and Georgeff, however. Georgeff's team is concerned with simulating the conscious cognitive processes of humans. His model of the robot is based on philosophical concepts of beliefs, goals, and intentions; Rosenschein's is based on computer science concepts of automata. Georgeff's is more "cognitive," Rosenschein's more "mechanistic."

"Stan starts with input-output circuits and then tries to string them together, whereas we build an architecture consisting of beliefs, goals, and intentions and try to express the robot's behavior in terms of these psychological attitudes."

Rosenschein's position is that psychological concepts like B-G-I's are of limited usefulness unless we can see ways of implementing them in computer programs, and that some of the ways these concepts have been implemented by AI researchers have resulted in systems that are too slow.

"Stan's idea is to build simple amoebas and to put them together to make complex machines," says Georgeff. "In our machines, on the other hand, you can literally take them apart and say, here's the system's beliefs, here's the system's intentions, and here's the goals. In Stan's it's hard to say where the beliefs are."

Rosenschein argues that his team is already developing designs for incorporating higher level reasoning. Georgeff says he can speed up lower level activity by compiling it or wiring it in later. Both will try out their systems on Flakey. Future SRI systems may involve a synthesis of the two approaches.

But, is it not presumptuous to think that a machine can really simulate the human conscious mind in all its complexity?

"Well, you can always simplify the problem or you can simplify the machine," Georgeff says. "To make an analogy, we'll be happy at this stage if we succeed in building a mechanical lizard. After that we may go on to build mechanical monkeys and more complex things." □

Natural language understanding

"John says that Mary believes that Sam doesn't want to go to Palo Alto"

Although natural language interfaces to present expert systems might be able to diagram the grammatical structure of sentences like this, they cannot really understand them or reason about their content.

"This is just not the kind of content or structure that expert systems tend to deal with," says Stan Rosenschein. But Flakey will need to reason about beliefs, goals, and intentions if he is to engage in intelligent communication in natural language with humans.

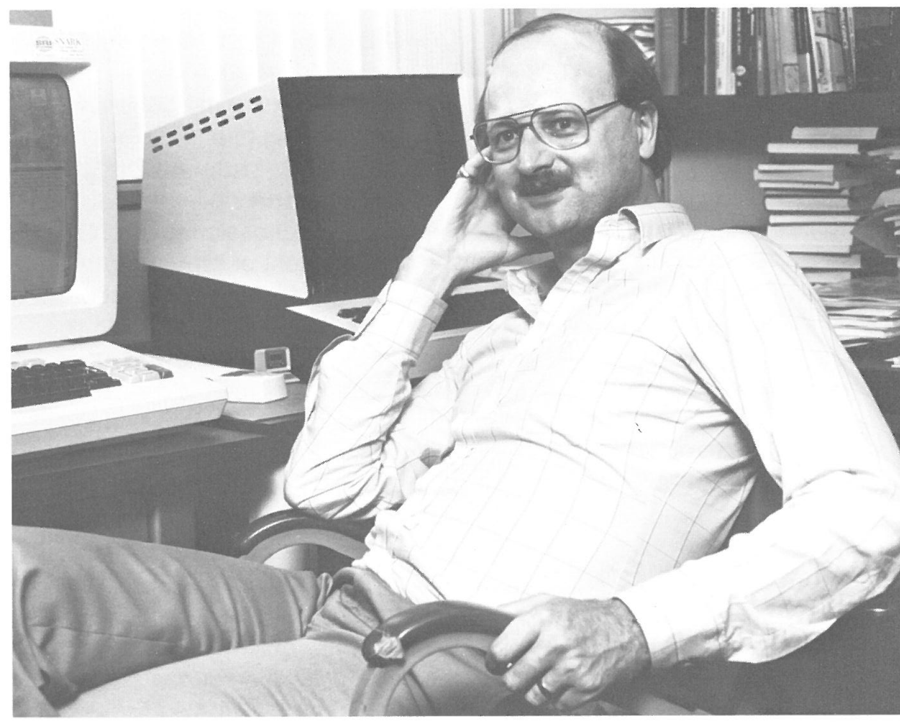
Designing computer systems that can use natural language intelligently is the ultimate goal of the Natural Language Program. To that end, the program is designing and testing experimental systems in which computers interpret English texts and use English to respond to instructions or answer questions addressed to databases and expert systems.

"Understanding" involves problem-solving and theorem-proving.

There is close collaboration between this program and the Reasoning and Representation Program, since both emphasize the use of formal logic. After all, as Program Director Ray Perrault points out, human languages serve primarily to represent and communicate knowledge. Moreover, when we hear a sentence, we must use reasoning in order to understand it. In doing this, we use our knowledge of the vocabulary and grammar of the language (its syntax) and we also use our knowledge of the world and of the immediate context in which the language is being used (semantics and pragmatics). "Understanding" involves problem-solving and theorem-proving (e.g., "All men are mortal. Socrates is a man; therefore Socrates is mortal.")

Logics

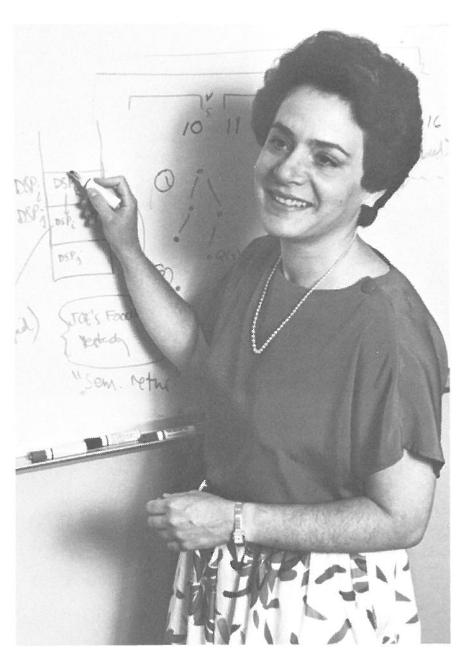
In programming a computer to understand natural language, knowledge of the vocabulary and syntax of a language is represented in a formal grammar that can be applied by the computer to "parse" a sentence and obtain its syntactic structures, showing how the words it contains are related to one another, as subjects and objects of verbs, for instance. These structures are translated into logical expressions that can be reasoned about, using the precise rules of formal logic together with axioms and concepts represented in a data or "knowledge" base. The analysis and interpretation of complex sentences like "John promised he would arrive on Monday" require reasoning with complex logics of tense and modality.



Ray Perrault

Several members of the program, including Lauri Karttunen, David Israel, and Jane Robinson, are working on formal grammars that provide the structures and expressions on which this kind of reasoning can be based. Some, including Fernando Pereira and Stuart Shieber, are developing formalisms for writing "direction-free" declarative grammars that can be used efficiently for "generating" sentences as well as for parsing them.

"When a grammar is developed primarily for parsing, you assume that you need to process a sentence from left to right," explains Perrault. "The computer would first process the subject, for example, determine whether it was singular or plural, then pick up the verb and simply check to see if it agreed with the subject. This works well for sentence-understanding. However, in sentence-generation, it's often more reasonable to first pick a suitable verb and then decide what the subject is going to be, and that means processing from right to left."



Barbara Grosz

Got into trouble

Fernando Pereira heads a project that uses a direction-free grammar in a natural-language knowledge-acquisition system called CANDIDE. "In the past, knowledge-acquisition systems have been oriented towards a question-and-answer format," says Pereira. "What's different about CANDIDE is that we not only ask questions but we also give new information to the system in English. We call the system CANDIDE because Voltaire's Candide took everything people told him very literally and then got into trouble, realizing later that people didn't quite mean what they seemed to be saying. We need learning systems that have some world-knowledge so that they can discriminate between

Continued next page

sense and nonsense. At present CANDIDE has very little world knowledge and can do only very simplistic checks of the coherence of what it's being told."

"...a sentence is viewed as an action that alters the mental state of the hearer..."

Speech is action

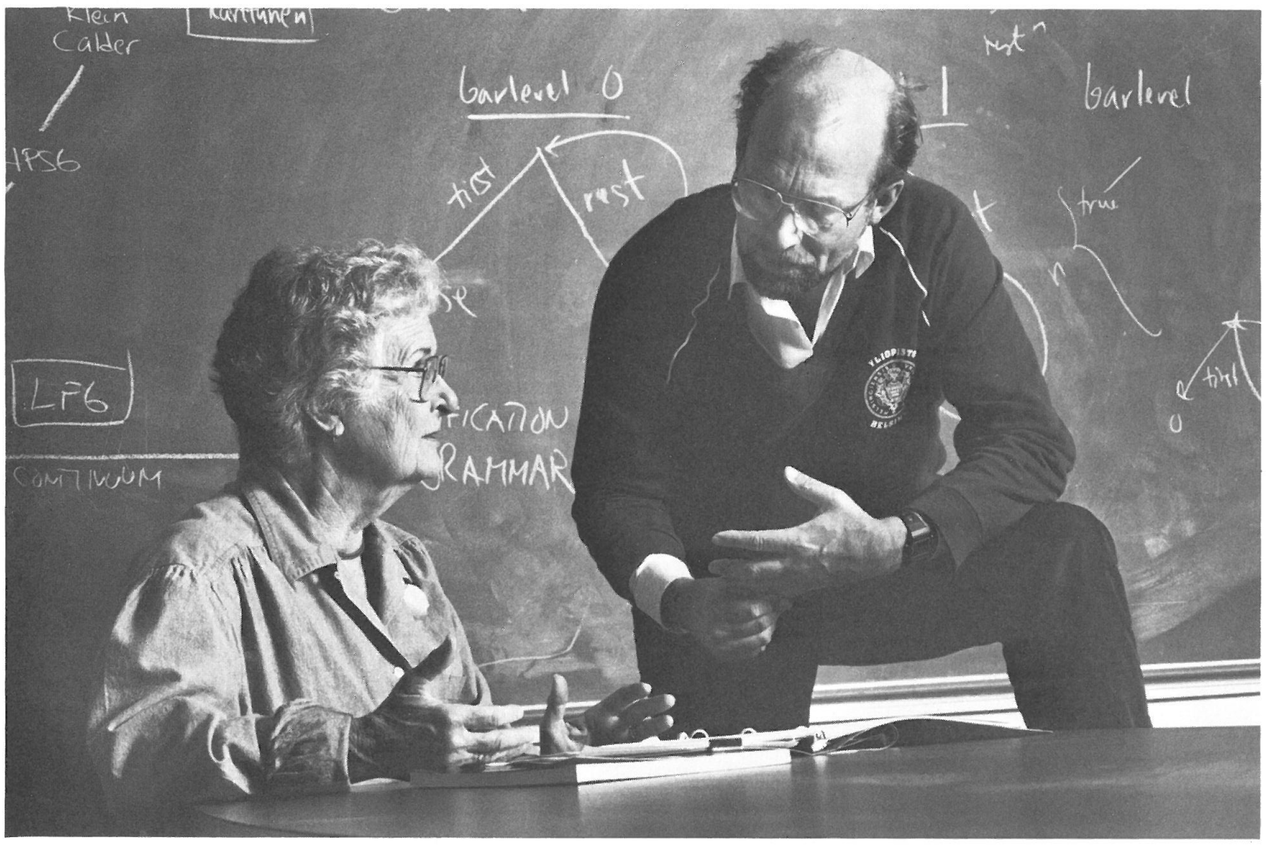
Another area of basic research is the development of "speech-act theory", carried on principally by Ray Perrault and Phil Cohen. In this theory, a sentence is viewed as an action that alters the mental state of the hearer, his beliefs, goals, and intentions. However, the syntactic form of a sentence and its literal meaning provide only partial clues to its contents (its semantic and pragmatic interpretation). "Can you lift the chair?" might mean any one of the following: "Will you please lift the chair?", "Have you the strength to lift the chair?", "Do you have permission to lift the chair?", or "Do you know how to lift the chair?". To interpret such a sentence in the context in which it is used, the hearer (receiver) must analyze it and reason about the possible beliefs, goals, and intentions of the speaker (producer). This activity on the hearer's part is a "recognizing action". To generate a sentence, a speaker engages in a "planning action", reasoning about how to express himself so that the sentence has the intended effect.

"These problems of language have interested philosophers and linguists for a long time, and computer science is bringing a whole set of new tools to their study," Perrault says. Some of these new tools are being put to use in a project headed by Doug Appelt, which involves interfacing a planning system with speech-act axioms and a declarative grammar to generate sentences that will satisfy the goals of a speaker by conveying his intended meanings effectively. The project also incorporates a new approach to theorem proving, developed by Mark Stickel. Jerry Hobbs is leading work on developing database models that underlie a domain of discourse and studying their relationship to the kind of commonsense reasoning that must be made available to a computer system that, unlike Voltaire's Candide, can use natural language appropriately.

Barbara Grosz, who recently left SRI for Harvard University's Computer Science Department, opened up a whole new area of study during her years with the Natural Language Program. She is developing a theory of how to deal computationally with the problems of machine interpretation of sequences of sentences in a coherent discourse. How does one know what a pronoun refers to, in a conversation made up of many sentences produced by different speakers? Often it refers to something mentioned many sentences back and not to the object referred to by a closely preceding noun. Grosz has shown how this and similar problems are related to the shifting of the focus of attention from one set of objects to another. Her computational models of how discourse structure reflects this shifting of attention is being further developed here by Martha Pollack.

Black art

How close is the group to achieving its goal of creating programs that can use natural language intelligently? No extravagant claims are being made. "All the systems we have at present," says Pereira, "tend to have areas where the processing is very much ad hoc, where it's like a black art. In our various systems, we are trying to develop a much more principled way of doing semantic interpretation and pragmatic processing. There are some solid principle of how such systems can be organized." □



Jane Robinson, left, and Lauri Kartunnen

Perception

A field of grass

How do you make a machine aware of its surroundings?

"You start one way or another with a machine being told what to expect," says Martin Fischler, who heads the Perception group. "It has to be given an understanding of the world around it."

A major project, funded by DARPA, is to create the technology for a vision system for a mobile robot. One such robot is the "autonomous land vehicle" built by Martin Marietta. It looks like a truck but is an automatic device and is to have the ability to reach a destination on its own. Eventually it will have all sorts of sensors to perform a range of tasks, but right now the main problem is navigation.

Like a human, the robotic vehicle sent on a trip across land will not want to fall into holes, will want to recognize what can support it and what can't, will want to recognize landmarks so it won't get lost, and will have to be able to follow instructions.

How does it recognize a field of grass, if the instruction says: "Turn right when you come to a field of grass"?

"We don't really have good tools for doing that," says Fischler. "The conventional mathematics and imaging methods are not the right languages for talking about natural scenes. And a stored image is not enough because you'll never see the same scene again in the same lighting."

"The conventional mathematics and imaging methods are not the right languages..."

What it should know

So part of the research is to find new languages for telling the machine what it should know about the environment, new languages for giving the machine internal models of the environment that it can use later as the basis for its interpretation of sensory evidence.

"Most of our work is theoretical, not in building sensors," says Fischler of his program. "Our main goal is to provide the technology for a machine that can interpret pictures or other sensory data. A major underlying problem is, how do you describe to the machine the nature of the things it sees in the world so that it can assign labels to them and understand them?"

Martin Fischler

Although the program's main focus is on fundamental issues in image-understanding and on applications such as robot vision and automated cartography, it also does significant work in other forms of extracting information from sensory data. For example, understanding the electromagnetic environment of a battlefield: determine, from the various signals received, what kinds of equipment are operating in the area; where radars or other kinds of emitting devices are located, what types are present, and how they are being used.

The work in image understanding is usually done on sensory evidence from imaging devices. These devices may not always produce photos but the equivalent of a photo. There may never even be a picture created on a piece of paper. The TV camera "eye" may feed directly into the machine, and the machine directly interpret the sensory input from that camera without the intermediary of a picture.

Already the autonomous land vehicle has been demonstrated navigating by itself along a simple road. Its "eyes" are typically a camera and a laser range-finder. Eventually it may be outfitted with other sensors like some acoustic or sonar devices for locating obstacles, and possibly infra-red sensors.

The laser range-finder is an imaging device that produces a "range image," as opposed to an "intensity image"—the pattern of light and dark intensities produced by a camera. "A range image is not a conventional image," says Fischler. "Instead of encoding the intensity at each point in the image, it encodes the distance from the viewer at each point. If you looked at it you might not understand what you were seeing. However it's a picture in the sense that the geometry of the scene is captured."

The image-understanding ("scene analysis") work calls for collaboration by computer scientists, physicists, mathematicians, psychologists, and electrical engineers. The initial interpretation of sensory data focuses on how light waves, sound waves or other kinds of electromagnetic radiation bounce off an object; that is, a problem in physics. Problems in mathematics, like those of differential geometry, also enter into the initial interpretation. The problem of data ("knowledge") representation and the need for new languages is perhaps the most basic and difficult problem, calling for collaboration by several disciplines. Then there is the problem of evaluation of sensory evidence, and here psychology also plays a role.

"Even if you have a good language to describe the scene, you can still get ambiguous information," says Fischler. "The sensors may make mistakes, and sometimes what you told the machine isn't completely correct, so there's the problem of evidential reasoning, or reasoning under uncertainty."

Continued next page

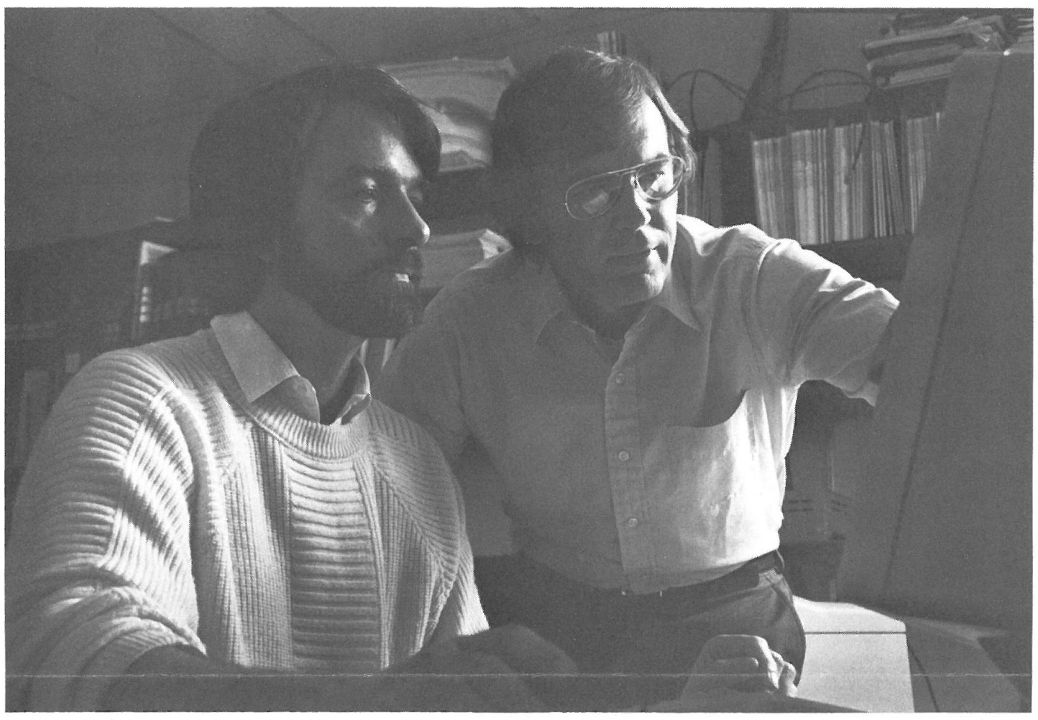
Hollywood helped
Has a language been found that can describe a natural scene? Not quite, but there are very promising new techniques, which have to be further developed. Some of them were pioneered in the movie industry, techniques involving "fractal textures" and "particle processes". They are special mathematical functions based on Benoit Mandelbrot's fractal geometry that produce very natural looking scenes.

"There are two parts to the problem of representing a natural scene in a language understandable to the computer," Fischler says. "You want a language that produces a very realistic looking picture, and you want the language very compact."

Fractal geometry is based on a kind of random statistical process. The texture of the surface in an image is a statistical variation in intensity, and fractal texturing enables one to capture the right intensity variations of a surface. Fractal representation is also very compact in that textures are described by a small number of distinguishing parameters.

To store a scene in the computer's memory, one would first take a "picture" of the scene. The image might be from a camera, a range-finder, a mechanical probe, or other kind of sensing device. Then one makes measurements on the image of the scene to extract the fractal textures of some of the objects in it. The information stored in the computer would not be an image but a set of parameters, a set of numbers, which is much more compact and useful. These parameters could then be used to re-generate the original scene, say, of a field of grass.

Important contributions to fractal techniques were made by a group in Marin County called Pixar Inc., formerly owned by Lucasfilms and recently bought by Steven Jobs. "They are mainly concerned with using the process to make realistic looking scenes synthetically," says Fischler, "but they haven't addressed the problem of how to make measurements on real scenes so that you can describe them or re-create them." His group's Sandy Pentland has been working on this latter problem.



Dave Marimont, left, and Bob Bolles

The real dialogue will not be between the two sides of the robot's brain but between two sets of robot designers.

Another very promising technique for deducing a 3-D scene description is called "epipolar plane image analysis," the work of the Perception Program's Robert Bolles, Harlyn Baker, and Dave Marimont. This technique first assembles and then analyzes a solid cube of data, made up of hundreds of pictures of a scene taken very close together. The pictures are taken with a regular camera moving in a straight line, looking to the side, and taking one picture after another every fraction of an inch. One axis of the space-time cube is the time axis. If one makes a cut perpendicular to the time axis one gets a conventional photo of the scene.

"If you cut this cube in an unconventional way you can get something that doesn't look like a normal picture but something that turns out to be much easier for interpreting the scene than a normal photo," says Fischler. "When we cut in unconventional ways the information is very structured, even though the real world may not be structured. Each point in the scene becomes a line in the new picture, and the slope of the line tells you how far the corresponding object is. From a single photo it would be very hard to deduce certain things, but because we're looking at parts of the scene over a period of time, we can integrate that information in a very nice way."

What is it to know
"Part of the motivation for Flakey is to stimulate research at the boundaries between disciplines," says Stan Rosenschein. "Although our people are mathematically oriented, they are sophisticated in different parts of mathematics. Geometry, differential equations, and continuous mathematics (like calculus) are the tools of the trade of the vision researchers, but mathematical logic and discrete mathematics (like combinatorics) are the tools of the reasoning and representation people. At a certain level they will have to connect up." The real dialogue will not be between the two sides of the robot's brain but between two sets of robot designers. Constructing programs is still more of an art than a science.

"At this point different programs don't mesh very well with one another because every program is handcrafted," says Rosenschein. One of the reasons is the lack of sufficiently precise answers to the question of what it is for a program to know anything in the first place. "What does it mean for a machine to have knowledge and to exploit knowledge? I feel that over the past few years there has been progress on foundational issues, and over the next ten, twenty years we'll see that translated into technology." □

Coming: *In the next issue, Inside SRI will look at the division's Network Information Services Center, Information Sciences and Technology Center, and Special Communications Systems Laboratory.*

by Heather Page

Cultivating and catching top talent for SRI's Engineering Research

"It's very competitive."

Michael Patrick is Manager of Group Human Resources, with responsibilities in HR systems, Engineering Research, IMEG, and the Washington, D.C., office. Before joining SRI in 1983, he was at Raychem Corporation. He has a bachelor's degree in urban studies from Ohio State University, and a master's degree in community planning from the University of Cincinnati.

Michael Patrick, shown with some of the 10,000 resumes he receives each year.

Q What are the basic requirements SRI's Engineering Group looks for in its applicants?

As with the rest of the Institute, the minimum standards tend to be rather high. There's a different process involved in evaluating people coming fresh out of university versus those who have a number of years of experience, but in both cases outstanding technical qualifications are a must. If the technical standards aren't met, we simply don't go any further. Other elements of the staffing formula include ability to work in a team, writing and speaking ability, initiative, and enthusiasm.

Q Is it hard to attract applicants?

SRI is easy to sell given what we do and how we go about doing it. My college recruiting schedule consists of research oriented schools like Berkeley, Stanford, MIT, Carnegie-Mellon, Illinois, Caltech and a few others. Students, particularly graduate students, have heard about us and have probably read some papers published by our staff. People are attracted to a place where they have the opportunity to interact with outstanding research staff, pursue their own research activities, and publish. In addition, for experienced people who may be coming from a large industrial organization, our smaller size is appealing. They can see the results of their contributions and feel like an important part of something rather than just another employee number.

Q Where do you get your applicants?

In addition to the schools I mentioned, we advertise in various publications. Also, one of our best sources is personal contacts of staff members. These can range from college friends to contacts through clients, competitors or professional associations. Then, of course, there are the general mail-ins.

Q What kind of applicants land the job?

In sum, I would say it's people whose backgrounds show a consistent pattern of exceptional results. This can range from simple grade averages, to types of research pursued, to working with diverse groups of people, to keeping up with, or in some cases producing, the "state of the art." We look for the well-rounded person who can engage in advanced research and please a client and then write about it all.

Q What's your biggest problem in recruiting?

Probably, it's dealing with the cost-of-living shock, particularly with regard to housing. It isn't so much a problem if someone is coming from Boston or New York, but it is if they're from North Carolina or New Mexico. Depending on the candidate's personal circumstances, we may tailor a program to attract him or her to the Bay Area.

Q How does your interviewing process work?

It's a filtering process. For universities, I'll interview students on campus and recommend certain ones to the relevant orgs for interviewing. I'll also work with the org on scheduling interviews for people responding to an ad. I'll screen out the respondents whose background doesn't match the job requirements. The orgs will identify who they want applications from and then who they want to bring in to interview. My rule of thumb is that the orgs are served well if they interview three to five people for each open position.

Q How is the competition and how do you fight it?

It's very competitive. Students from the better schools like MIT have 2 or 3 job offers in addition to an SRI offer. We tend to do fairly well even when we're competing against the larger "name" companies, and we don't lose people because of salary. We've always made sure we're very competitive in that aspect. We sometimes lose candidates because of some other reason, like perceived greater opportunities for advancement elsewhere.

Q How many applications do you get a year?

Between advertising, college recruiting, personal contacts and cold mail-ins, we go through close to 10,000 a year. As a way of keeping track of all these contacts, I've set up a database on my PC. I use it for several things—mailings, an information file, and a tracking system. Even if we don't have a job for them right now, it's a way of keeping track of them when something in the future comes up. □

Notables

Jack Goldberg, senior staff scientist, Computer Science Lab., was elected a 1986 Fellow of the Institute of Electrical and Electronic Engineers (IEEE). Goldberg received this honor for his leadership in fault-tolerant computer development programs. He is also editor of the IEEE Transactions on Software Engineering.

Luther "Bud" Smithson, director, Biotechnology Program, IMEG, was quoted in an October Christian Science Monitor article, "Genetic Engineering: Part 3/The Business." Smithson states that many biotechnology companies have established overly ambitious research projects. "The Eureka-type products that people expected have not occurred," he said.

SRI President **William F. Miller** has been elected a fellow of the American Association For the Advancement of Science (AAAS) for "distinguished contributions to research, teaching, and scientific administration."

Gerry Andeen, staff scientist, Mechanical Research Lab., was elected to represent Ward 6 of the Midpeninsula Regional Open Space District. Andeen, former mayor and councilman in Menlo Park, is responsible for managing more than 20,000 acres of public open space in San Mateo County.

"SRI—The Take-Off Days," a new book by Senior Director **Weldon B. Gibson**, picks up the story of the Institute's history where his earlier book, "SRI—The Founding Years," left off. The second book focuses on developments during the decade that followed the creation of the Institute in 1946 and ends with the mid-1950s. By then, according to Gibson, SRI had achieved "a certain maturity, critical mass and self-sufficiency."

Self-sufficiency in advanced computer science is fast becoming a major strategic resource, according to "Intelligent Machinery: Theory and Practice," edited by **Ian Benson**, consultant, SRI Croydon. The book focuses on medium and long-term directions in software research, and the role of national programs in the United States, Japan, and Europe. The book is a product of a conference organized by SRI which reviewed the research agendas of the major national programs to foster so-called Fifth Generation computers and software.

Wanted: *Items for our Notables column. Please send in news about SRI staffers' activities, achievements and awards.*

Inside SRI

Inside SRI is published four times a year for SRI employees by the Office of Corporate Communications.

Editor Julie L. Wei
Designer Ronald Moore
Photographer Damian Conrad

Editorial office: Room AA-220, SRI International, 333 Ravenswood Ave., Menlo Park, CA. 94025. Telephone: (415) 859-4787. Telex: 334486. E-Mail: Wei @ sri-kl

Pictured with the winning trophy are: (1) Joe Grippo, (2) James Bupp, (3) John Ryan, (4) Steve Rooks, (5) Doug Bamford, (6) Dave Austin, (7) Paul McKenney, (8) Sue Romano, (9) Mariano Caunday, (10) Linda Huber, (11) David Sands, (12) Rich Llewellyn, (13) Don Shockey, (14) Co-Captain George Black, (15) Lori Kanerva, (16) Co-Captain Gordon Bliss, (17) Brock Hinzmann, (18) Ed Claasen, (19) Howard Fisher, (20) SRI President William F. Miller, and (21) Kay Donnelly. Not pictured: Susan Lohrer and Anne Smith.

Our champion track team

SRI's Track Team took first place among companies with less than 5,000 employees in the U.S. Corporate Track Association (USCTA) National Championship races, held in Los Angeles, July 26-27. SRI placed first or second in eight out of the nine races contested and won the meet by a wide margin over second place Aerospace Corporation of Los Angeles and third place EPRI of Palo Alto.