conduit!

Volume 3, Number 2 — Department of Computer Science, Brown University — Fall, 1994

KAELBLING RECEIVES PRESIDENTIAL FACULTY FELLOW AWARD

Leslie Pack Kaelbling

The National Science Foundation's Presidential Faculty Fellow program seeks to recognize early in their careers the scholarly achievements and potential of the nation's most outstanding science and engineering faculty members. It identifies potential leaders in academic endeavors including research, teaching, and administration. We are impressed and delighted that the President, in conjunction with the National Science Foundation, has selected our colleague Leslie Pack Kaelbling as a 1994 recipient of this highly prestigious award. Leslie was one of the 30 outstanding faculty selected from a field of 270 nominees. It is interesting to note that several of the awardees were NSF Presidential Young Investigators—Leslie received this distinction in 1992. Leslie, who received her A.B. and Ph.D. degrees from Stanford, has been an Assistant Professor in this department since 1991. The $100,000 she will receive per year for three years will help fund the work of three Ph.D. students assisting Kaelbling with her research. The Presidential Faculty Fellow award will be presented by President Clinton in an upcoming White House ceremony.

Kaelbling's research focuses on the design and implementation of autonomous agents, the aim being to develop program specification techniques, planning algorithms and learning methods to make possible the design and construction of very sophisticated autonomous agents by greatly easing the burden on the human designer. "You can't learn if you know nothing at all," says Kaelbling. "The challenge is to get robots to take whatever knowledge programmers have installed in the initial structure and to build on it, so we're trying to invent learning algorithms through which robots can make decisions based on previous experience and prior knowledge. Even if a robot learns something very simple, it's a basis from which to grow." For the purpose of this work, an autonomous agent is taken to be any computer program that has a sustained interaction with a dynamic, incompletely predictable environment. Although mobile robots have been used as a canonical agent type, it is also possible to consider agents that control factories or manage database consistency.

> *"You can't learn if you know nothing at all.......Even if a robot learns something very simple, it's a basis from which to grow"*

One way in which programs for agents differ from computer programs in general is that they must operate in a dynamic and unpredictable environment. Operating in a dynamic environment requires the agent to choose its actions quickly so that they are still appropriate to the state of the environment that was most recently observed. Operating in an unpredictable environment requires the agent to monitor the state of the environment continuously and prevents it from planning actions far into the future.

Leslie's research at SRI International, where she worked while a graduate student, focused on methods of constructing programs for autonomous agents. The first step was to define formal semantics for programs embedded in a dynamic environment. Together with Stanley Rosenschein she developed situated-automata theory which allows the designer of an agent program to describe the semantics of that program in terms of correlations between its internal state and the state of the environment. With the theoretical foundations in place, they set out to design programming tools to facilitate the implementation of agent programs. Although formal, symbolic structures are a good medium for humans to use to express interactions between the agent and the environment, they are too computationally complex to manipulate on-line in an agent. They developed two languages—Rex and Gapps—that let the programmer use symbolic specifications of agent behavior but compile into very efficient programs for the agent and guarantee real-time reactions. Leslie implemented compilers for Rex and Gapps that were used extensively in programming the SRI mobile robot, Flakey, and have also been used by researchers in other laboratories.

"it would be more efficient and effective for the agent, rather than the programmer, to do the learning about its interaction with the environment"

Even with programming tools that make it easy for programmers to specify what they know about how the agent should interact with its environment, the programming problem is still very difficult and requires many debugging iterations. From much experience of this process, Leslie learned that this difficulty arises because programmers rarely actually know the correct behavior for the agent. Even with specifications of how a given set of sensors and effectors works, it is nearly impossible to write a program that uses them correctly, and the programmer ends up learning a great deal, during the debugging process, about how those sensors and effectors interact with the environment. As a result of this observation, she became convinced that it would be more efficient and effective for the agent, rather than the programmer, to do the learning about its interaction with the environment.

Leslie's Ph.D. thesis research, carried out at Teleos Research, concerned algorithms for learning from experience. This problem, called *reinforcement learning*, differs from the more frequently considered problem of concept (or function) learning. In both cases, the object is to learn a mapping from an input space to an output space. In the concept-learning framework, the learning algorithm is given a set of input-output pairs from which to learn the function; in reinforcement learning, it is assumed that the program is in control of an agent. The agent gets an input (a sensory reflection of the current state of the world) and chooses an action to take in the world. The agent then receives a reinforcement value that indicates how good the new state of the world is for the agent. The agent must learn a mapping from states to actions that maximizes reinforcement. Crucially, though, the world does not tell the agent which actions to take in which situations; the agent must discover this through a considered strategy of trial and error. Leslie's thesis work consisted of a number of novel algorithms based on statistical methods and extensive empirical comparisons between her algorithms and existing methods from learning-automata theory and neural networks.

In the course of this work, Leslie identified some important areas for future work, some of which she is now pursuing. Her plan for future research is to continue to explore methods for reinforcement learning and to reintegrate learning methods with off-line specification. She will continue to focus on the design and implementation of autonomous agents. A strong area of emphasis will be integration—to understand how to integrate learning methods with pre-programmed strategies and how to integrate low-level, reactive methods with higher-level planning approaches. In addition, she will continue to explore some technical problems in reinforcement learning.

Although Leslie has been studying reinforcement learning in the *tabula rasa* case, that is, when the agent has little or no prior information about the world, she did not find this to be the best strategy in the long run. In order to design and build useful agents, one will have to supply a great deal of innate knowledge and structure, then allow the agent to learn to refine and extend its knowledge. An important aspect of her future research will be to investigate this question, including what form the innate knowledge should take, how it can be specified by a programmer, and how the learning techniques will make use of it.

Recently Leslie has begun work on the problem of planning efficiently in stochastic domains. These planning techniques rest on the same formal foundations as the reinforcement techniques, those of Markov process theory. This shared basis makes it easy to integrate high-level planning and model learning with low-level reactive strategies and reinforcement learning. As more complex intelligent systems are constructed, it will be important to have well-integrated mechanisms working at both of these levels. She intends to develop the planning techniques further, to design a strategy for integrating the high and low levels, and to study what components of the agent's behavior are more appropriately specified at which level. Optimal solutions to the reinforcement-learning problem exist in the statistics and dynamic-programming literature. These solutions all require that the input and output spaces be completely enumerated and all have time complexities at least polynomial in the sizes of these spaces. For any interesting autonomous agent, however, the input and output spaces are very large, making these approaches infeasible. Her most recent work has begun to address this issue by developing algorithms that learn efficiently through hierarchical approximations of the state space. She will continue the study of these approximations with a student and investigate other methods of learning efficiently through approximation.

"My fundamental aim is not to make robots do a particular thing, but to understand ***how*** *we can get them to do a wide variety of things"*

It is rare that an agent has immediate sensory access to all of the relevant aspects of the world, yet nearly all reinforcement-learning work done so far makes this assumption. The problem of learning action strategies that have state is related to that of learning stochastic automata. Leslie has begun exploring this problem in another context, and intends to apply or extend these results to the problem of learning to track hidden state. With two graduate students, she has begun investigating the application of operations-research techniques of partially observable Markov processes for finding optimal strategies in problems with hidden state. These techniques are computationally intractable, but she proposes to investigate approximate techniques that will let them be used in practical learning systems. Progress in these areas will have substantial impact on the construction of intelligent autonomous agents.

Leslie has taken a five-year detour from robotics to study learning and will now have the opportunity to test her findings and hypotheses. "I hope to find the answers," she says. "My fundamental aim is not to make robots do a particular thing, but to understand *how* we can get them to do a wide variety of things."

The final exam for Professor Kaelbling's robotics course was a robot talent competition!

Leslie is a gifted and creative teacher, earning recognition for excellence in student teaching at Stanford and playing a leading role at Brown in redesigning the undergraduate CS curriculum, among other teaching accomplishments. Recently, she introduced a popular robot-building course using Lego blocks, detailed in last spring's issue of ***conduit!***, and designed a new introductory course in computer systems with emphasis on integration of hardware and software. Working closely with graduate students, giving them direction and learning from them, is key to Leslie's academic agenda. The Presidential Faculty Fellow award recognizes her significant achievements as a researcher and teacher and her potential for future contributions to the scientific and educational enterprise of the nation.

RESEARCH AFTER THE COLD WAR

John Savage

The unwritten social contract that has existed since 1948 between the U.S. government and the American research community has expired with the end of the Cold War. Research can no longer be justified by its potential contribution to national defense; now, its economic potential is paramount. This far-reaching shift in the nation's priorities is reflected in recent reports by the Carnegie Commission on Science, Technology and Government, the National Academies of Science and Engineering, the Institute of Medicine, Congress's Office of Technology Assessment and several congressional committees.

In this changed climate, and noting the high cost of the multi-billion-dollar Superconducting Supercollider and other basic research projects, many government and business leaders are demanding that federal agencies adjust their funding to the new economic imperatives. These demands alarm many academics, and naturally so. They do not want their knowledge and skills to go to waste or their pet research projects to go unfunded; but because there is an element of self-interest involved, their genuine concerns about the nation's shift in priorities are too often dismissed as merely special pleading. This is very unfortunate for the nation.

"the 'golden eggs' that we all prize so much, such as the computer.....will stop appearing unless we maintain our national commitment to the 'goose'—basic research"

America has a very practical interest in basic research, even if most Americans, with our characteristic and sometimes excessive emphasis on the practical, may not realize that. The truth is that the "golden eggs" that we all prize so much, such as the computer and communications satellites, will stop appearing unless we maintain our national commitment to the "goose"—basic research.

There has been a fundamental change in the nature of basic scientific research since the 1950s. Until fairly recently, fundamental research was conducted with relatively simple equipment. Galileo and Copernicus studied the planets and stars with telescopes; Mendel cultivated his garden with hand tools. In the 19th century, Faraday conducted experiments with batteries, wires and magnets. In the 20th century, Michelson and Morley built an interferometer with mirrors, and Eckert and Mauchly built a programmable electronic computer with vacuum tubes. Even today, the theoretical models and methods of analysis that explain these results can generally be understood by undergraduate students.

However, science and engineering changed dramatically during the Cold War. Research and development now deal with complicated systems for which compact models are either non-existent or inadequate. As a result, much modern research cannot be accomplished without long computer simulations on the most powerful machines available. Unfortunately, since these machines are very expensive, they must be shared, thereby limiting the time available for any one simulation. Moreover, simulation often must be done for each small variation in a problem, which means vast numbers of simulations for the same fundamental problem.

Today, computer simulation is essential for learning about phenomena that are difficult to understand through experiment or mathematical analysis. But simulation's drawbacks make it desirable to learn how to achieve the same ends with less computer time. In the short term, we need improved computers, algorithms and methods of analysis to simulate with known techniques. In the long term, we must develop new models and methods of analysis that make possible predictions of the behavior of complex systems with far less simulation. This should be a major focus of long-term basic research.

American science and engineering are now poised to make major contributions to the economy. Advances in materials science make possible new lightweight, strong, energy-efficient products. Genetic engineering offers hope of curing important inherited diseases. Molecular modeling is simplifying the design of new drugs. Simulation of quantum-mechanical systems helps in designing chemical reactions and improves understanding of particle physics. Computational fluid mechanics is leading to the design of more efficient aircraft. Developments in computer and communications technology are altering the ways people work and enhancing their productivity.

Basic research is a high-risk, long-term activity. Very few projects result in a major economic impact. Since it is very difficult to predict winners and losers, many researchers need to be supported. No one could have anticipated that three Bell Labs researchers who were looking for an improved vacuum tube would end up discovering the transistor. Nor could anyone have foreseen that two academics seeking to improve the computation of ballistics firing tables would wind up inventing the programmable electronic computer. These two inventions have brought us, and our economy, into the computer age.

As a nation, we can support research and development efforts that have obvious commercial and industrial applications, but if we ignore the basic research that does not have those obvious applications, we may fail to find solutions to the problems that we agree need to be solved. The Cold War is over, and it is clear that we need to adjust our research priorities accordingly; but if we focus too much on the seemingly practical, we may find, in the end, that we have been extremely impractical. The losers may include not just the researchers but the welfare of the nation.

THE 13TH IPP SYMPOSIUM

Tom Doeppner

A well attended, lively IPP Symposium was held April 26 on the topic of *Mobile and Ubiquitous Computing*. For this hot area, a number of leading researchers and practitioners from industry came to discuss their work.

Leading off the day was Mark Weiser, of Xerox PARC, who spoke on *Ubiquitous Computing*, a field he founded. His work has been written about widely in the past few years in such popular magazines as *Scientific American* and *Smithsonian*. Besides giving an excellent presentation, he was the first IPP speaker to relate her or his work to radical feminism. Weiser demonstrated mobile computing by communicating with me via cellular email while travelling from the west coast to the east coast.

> **Weiser's abstract**: Inspired by the social scientists, philosophers, and anthropologists at PARC, we have been trying to take a radical look at what computing and networking ought to be like. We believe that people live through their practices and tacit knowledge so that the most powerful things are those that are effectively invisible in use. This is a challenge that affects all of computer science. Our preliminary approach: Activate the world. Provide hundreds of wireless computing devices per person per office, of all scales (from 1" displays to wall-sized). This has required new work in operating systems, user interfaces, networks, wireless, displays, and many other areas. We call our work "ubiquitous computing." This is different from PDAs, dynabooks, or information at your fingertips. It is invisible, everywhere computing that does not live on a personal device of any sort, but is in the woodwork everywhere. I describe the origins of ubiquitous computing in post-modernist thought, why ubiquitous computing is inevitable, and some of the things we have learned from our research.

The next talk was on *Technical Challenges in Mobile Computing*, by David Oran of DEC. Oran was widely hailed within DEC as their architect for mobile computing. He did not disappoint, giving a talk that was considered one of the best by the attendees.

> **Oran's abstract**: Discussions around the technologies involved in mobile

computing often get narrowly focused in one of two areas: either the design of the portable device itself (PDA, tablet, etc.), or the deployment and idiosyncrasies of wireless networks (Mobitex, CDPD, etc.). While each of these areas represent important aspects of the whole problem and provide many opportunities for innovation and customer value, the success of mobile computing in providing customer solutions to business problems goes well beyond the wireless medium and the portable computer.

Symposium speakers: back row, l to r, David Oran, Charles Perkins, Clem Cole; front row, l to r, Tom Doeppner, Mark Weiser, Bob Frankston

This talk examines some of the system-level aspects of mobile computing and highlights the design issues faced by customers wishing to deploy such systems. I address the following issues:

1. Providing robust communication over multiple networks with wireless low-bandwidth links attached to high-speed backbones 2. Providing global roaming for users, not just geographically within the coverage area of one service, but among service suppliers and different wireless (and wired) technologies. 3. Dealing with intermittent connectivity and shielding the user, and in some cases the application, from periods of disconnection. 4. Providing end-to-end security beyond simple protection from eavesdropping on wireless links. The talk concludes with some proposals for what the ideal system would look like and what user expectations can and cannot be met by technologies available today and in the near future.

Clem Cole, of Locus Computing Corporation, presented a talk on *Computers Without Networks*, which was an overview of the area of mobile computing. Cole had recently chaired a Usenix symposium on the subject and was well versed in what is going on.

Cole's abstract: With this talk we will examine the current trend of computing that is not 'bound' to physical media–Ethernets, token rings and the like. As the trend is now towards laptop, palmtop, and even PDAs, the computing infrastructure must rethink and redeploy to exploit this new generation of mobile computing devices.

Charlie Perkins, of IBM's T. J. Watson Research Center, spoke on *Mobile Networking at the IBM T. J. Watson Research Center.* This talk focused on adapting TCP/IP for mobile computing. Perkins has been one of the most important players in this area and has been a leader in the associated standards work. His talk was appreciated for being very down to earth.

Perkins' abstract: For several years we have been involved with developing protocols that can provide seamless connectivity for mobile users using TCP/IP. Numerous approaches have been tried, with varying degrees of success. In addition, we have been intimately involved with the process of creating a standard mobile-IP protocol document with the Internet Engineering Task Force (IETF) Working Group on mobile-IP. As part of that effort, many different approaches for mobile networking have been proposed, evaluated, and mostly rejected; however, valuable lessons have been learned from each new approach. I will try to put our work in perspective with these other approaches, past and present, as well as describing some implementation experience.

In addition, we have created and begun to evaluate a new scheme for ad-hoc networking, and have some results for real-time network management for mobile systems. I will describe our results in those areas and mention the joint projects we have begun with other academic institutions, as well as other related interesting projects within our Mobile Systems group.

Finally we heard from Bob Frankston, of Microsoft, who is best known for being one of the inventors of VisiCalc. He spoke on *Mobile and Ubiquitous Computing: Changing the Nature of Computing*. Besides giving an excellent talk, Frankston was the first IPP speaker to project his talk from a laptop computer, after making a few slides "in real time" that responded to points brought up in the other talks.

> **Frankston's abstract**: Ubiquitous computing represents a disappearance of computing into the "woodwork" or infrastructure. But the concepts from computing are very relevant. The issue isn't one of simply smarter devices or protocols, but touches upon really hard problems such as how do we allow people to add intelligence to many mechanisms (or, to use a better term, specify a policy) and yet keep the complexity of even modest interactions between simple systems from becoming overwhelming. Ubiquitous computing means getting beyond the naive models of computers as perfect engines and dealing with the awkward issues of fundamental ambiguities and imperfections in the real world.
>
> Changing the nature of computing means more than simply extending a protocol such as IP to mobile devices; it means reexamining the rationale behind these protocols in light of the new environment. Just as the PC wasn't a mini-mini (though it is becoming one), the devices in a world of ubiquitous computing are more identified with their application than with computing as such.

A great deal of discussion was generated by all of the talks. More than a few members of the audience probably deserve credit as speakers, since they contributed greatly to the success of the day.

ALUMNI EMAIL TO THE EDITOR

Keep it coming! sjh@cs.brown.edu

JIM GARVIN '78

I just received my latest copy of ***Conduit!*** (Spring 1994) and was heartened to see that some of the alumni email letters were from old friends of mine from those early halcyon days of CS at Brown.

My name is Jim Garvin, and I was perhaps a little-known CS department undergraduate who was fortunate enough to work for both Andy van Dam and Bob Sedgewick in the late 1970s (I graduated in 1978). Indeed, I count it amongst my greatest educational experiences to have studied the art of assembly-language programming under Andy in the now-extinct AM101 (a virtual marathon of a semester course back then) and the wonders of mathematical analysis of algorithms under Bob Sedgewick. Further, I was lucky enough to have served as a TA for both Andy and Sedge for several courses (I may have learned more by TA'ing for these great folks than in many of my classes!)

Now I am happily exploring the wonders of the Solar System here at NASA's Goddard Space Flight Center as a staff scientist, and thanks to my background in computer science and mathematics as an undergraduate at Brown, I am endeavoring to develop new approaches for investigating the processes which shape planetary surfaces. After leaving Brown in 1978, I spent a fruitful year at Stanford, where I completed my M.Sc. in CS under John Hennessy. During this exciting time I came to the perhaps late realization that I had an incurable passion for earth and planetary science, and thus returned to Brown in late 1979 as the final student of Dr. T. A. Mutch to pursue quantitative geomorphology studies of Mars and Venus.

I finished my Ph.D. at Brown in 1984 under Prof. Jim Head, and then was fortunate enough to land a staff scientist position here at NASA's Goddard Space Flight Center. In my nine years here I have spearheaded efforts to develop laser altimeter instruments to measure the shapes of planetary landscapes at unprecedented resolution (for Earth and Mars), and have studied the remote sensing signatures of impact craters around the Earth. Of course, I am still an avid APL programmer, in spite of all my good training at Brown CS, and am now finally learning object-oriented programming to enable me to develop the algorithms needed to interpret the huge database of aircraft laser altimeter measurements we have collected for volcanoes and glaciers around the planet, in anticipation of such measurements for Mars in a few years.

I would welcome contact with CS alumni—hope Brown CS Department continues to flourish in the roaring ’90s.

garvin@denali.gsfc.nasa.gov
301-286-6565

RUSS ELLSWORTH ’79, Sc.M. ’85

Russ Ellsworth

While driving home from my 15th reunion (which was a bit of a drag because that’s the year when hardly anyone shows up), there was this thought in the back of my mind which I couldn’t quite figure out. How come, after almost twenty years of being connected to the CS Department at Brown University, MY hair is getting sparse, I’m getting flabby, and my eyesight is getting worse, yet ANDY, PETER, JOHN, and TOM haven’t aged a SINGLE DAY? Is this what being in the field of education will do for me? I mean really, I first watched Andy teaching AM100 in February of 1975 when I was being recruited by the track team, and others I met within a year, and they’re still the same!

rellsworth@rcnvms.rcn.mass.edu

FACULTY RESPONSES:

John Savage

“In my case, the explanation is good genes!”

Andy van Dam

“Russ has proved that his eyesight is indeed getting worse—he now sees us through rose-colored glasses....”

Peter Wegner

“Russ, you obviously want something from us by flattering us like this. It is remarks like yours that keep us feeling young and make our work worthwhile.”

Tom Doeppner

“We must have looked pretty old to Russ when he was a student...”

ROSS KNIGHTS ’84

I have just started work at Apple Computer in Cambridge, Massachusetts, as Software Pathologist (i.e., quality assurance engineer) on the Dylan project. We are designing a highly interactive development environment built around a new dynamic object-oriented programming language. Prior to this I worked for almost seven years at Interleaf, in Waltham, Massachusetts, developing automated performance-testing and regression-testing software in Lisp.

knights@cambridge.apple.com

The Computer Science faculty in 1979. l to r, Tom Doeppner, Bob Sedgewick, Peter Wegner, Andy van Dam, John Savage, Eugene Charniak, Steve Reiss

Thomas Dean. Tom was one of 20 individuals elected as fellows of AAAI this year—the award was presented at the National Conference in Seattle in August. The book he has been writing these past two years with James Allen and John Aloimonos, *Artificial Intelligence: Theory and Practice*, will be in print in late October. Tom is currently on a leave of absence at the University of Washington.

Thomas Doeppner. Tom gave an invited talk at the DCE Developers Conference in Boston in August.

John Hughes. Spike's response to the request for activities copy read as follows: "My CS activities for the last six months are pretty mundane, I'm afraid. I taught two courses, spent some of the summer doing and directing research, attended SIGGRAPH in Orlando, and am about to have my first Ph.D. student graduate on Friday. The cause of this relative lack of interesting things is, of course, my new daughter Meg, but she's hardly fodder for ***conduit!***..."

Paris Kanellakis. Paris has been developing a new undergraduate course on Models of Computation that uses the Scheme language to illustrate concepts from the theory of algorithms and complexity.

Leslie Kaelbling. Leslie has given five invited talks at various national and international conferences since last spring, most notably at the European Conference on AI in Amsterdam and the National Conference on AI in Seattle. She is currently a member of several important committees—the Advisory Committee of the International Joint Conference on Artificial Intelligence, the editorial board for *Machine Learning Journal*, and the program committee for IJCAI, AAAI, Machine Learning Conference, Simulation of Adaptive Behavior Conference, and the International Conference on Planning Systems.

Pascal Van Hentenryck. Pascal is a member of three program committees this semester—SSA (Symposium on Static Analysis), PASCO (Symposium on Parallel Symbolic Computation), and IJCAI (area chair for constraint satisfaction). He is also an invited speaker at the University of Maryland. To prepare his computer architecture class, Pascal spent his time in August designing a simple pipelined RISC that he proudly displays to anyone interested.

John Savage. John gave an invited talk at MIT in the SuperTech Seminar Series in LCS and a paper at the 6th Annual Symposium on Parallel Algorithms and Architectures. His research is now directed toward algorithms and architectures for high-performance computing. He is on sabbatical leave this semester and is writing his third book, *Applied Theory of Computation*.

Roberto Tamassia. Together with I.G. Tollis, Roberto organized and co-chaired Graph Drawing '94 (DIMACS Workshop) in Princeton—more than 120 participants attended the October meeting. He also lectured on Graph Visualization with I.F. Cruz at the IEEE Symposium on Visual Languages in St. Louis.

Peter Wegner. In August Peter co-organized a workshop at Dagstuhl Castle in Germany, bringing together researchers in object-oriented languages, software engineering, databases, and theory. He is organizing a workshop on the introductory computer science curriculum at Harvard in January. Peter and Marvin Israel of Bellcore have been appointed Editors-in-Chief of *Computing Surveys.* They plan to solicit tutorials and surveys in emerging areas of computing—network interfaces, collaborative work, novel architectures, distributed system technology, multimedia and virtual reality systems, realtime planning and scheduling, etc. If you have any ideas for tutorial or survey articles, please get in touch with Peter (pw@cs.brown.edu).

Stanley Zdonik. Stan taught a major part of a five-day summer school course at MIT. His three-year ARPA contract on Object-Oriented Query Processing was extended for a fourth year, and he has received a gift from Intel Corporation to support work on broadcast disks for asymmetric network environments.

IT'S A ROBOT !

Name:	RAMONA
Delivery:	June 30, 1994, 2:30pm
	No complications
Weight:	240 lbs
Length:	48 ins

Ramona is a new RWI robot which will be used by Leslie Kaelbling and department students to investigate robot navigation and machine learning. Current robots can reliably cope only with environments which are highly structured, such as auto assembly plants, or highly modified, such as warehouses with magnetic tracks embedded in the floor. The goal of this project is to enable Ramona reliably to navigate in an unmodified and unstructured environment. In order to test this ability, Ramona will eventually serve as a messenger in the Computer Science Department, receiving requests for package delivery via email and carrying them out.

Ramona runs off an onboard battery system and can roam around the department at speeds of up to three miles per hour for several hours before requiring a recharge. She senses the environment through an array of 24 sonar sensors and two cameras on a pan/tilt platform. Computation is provided by an onboard network of three computers running Linux, which can be accessed via the departmental network or the laptop computer on the robot. Still in the works at RWI is a set of 48 infrared sensors and an arm which will allow Ramona to pick up packages and press elevator buttons in order to make deliveries on any floor.

Since July, students have completed the low-level software necessary to program Ramona to perform more complicated tasks. Ramona is currently able to navigate the corridors of the department and is beginning to learn maps of the environment. Leslie and her students hope that eventually Ramona will be able to deliver packages within the department without supervision.

FROM THE CHAIRMAN, Eugene Charniak

When the department was started back in 1978-79, John Savage thought it should have a defining philosophy; he suggested what has become a touchstone here, that “combining theory and practice” was the way to go. I think of this immediately in conjunction with the newest addition to our faculty, Maurice Herlihy, whom we are delighted to welcome this semester. Maurice’s work is in the general area of distributed computing. While he has worked in many subareas therein, he is perhaps best known for his work on concurrency and, in particular, wait-free synchronization. Maurice is one of the few computer scientists who is a theorist’s theorist and a practitioner’s practitioner. (The reverse is more common, regrettably.) While Maurice was officially hired to fill a “theory” slot in the departmental roster, one of our systems-building types commented that he knew of a well-known practitioner who published in the standard places for practitioners and had the same name and institutional affiliation! Maurice has another trait noteworthy in this context—his mother is a professor

Newly hatched Ph.D.s: l to r, Jak Kirman, Sai Subramanian, David Langworthy and Kate Sanders. Absent but not forgotten, Ted Camus and poultry!

Maurice Herlihy

here at Brown, in the department of history, as was his father until he passed away a few years ago.

We are proud that one of our graduate students, Jonathan Monsarrat, has won one of Brown's four annual awards to outstanding graduate teaching assistants. Jonathan won this award for his ability both to help students in the class and generally to inspire them to continue with their CS studies. To put it another way, Jonathan works hard to make a computer science student's experience at Brown a happy one, and his techniques range from the tried and true to the definitely unconventional. As an example of the former, he has had a large part in the development of the undergraduate robotics group, which has sent robots to compete in two of the three robot competitions run by the AAAI (American Association for Artificial Intelligence). Their robots didn't win, but they were the only undergraduate group to compete. Jonathan's less traditional techniques will go down in Brown Computer Science lore. For example, his thesis advisor is Tom Dean, who started the robotics group. One night Jonathan bought a bottle of nitrogen and a boxload of balloons, and he and some undergraduate volunteers filled Tom's office with the balloons (see picture). This particular caper has also gone down in university annals since, as it happened, the next morning university videotapers were following Jennet Kirschenbaum around as part of a video piece they were doing on

Tom Dean's office—enough said!

winners of the Brown Says Thank You Award for distinguished staff service (see ***conduit!*** Volume 2, Number 2). One of the things that landed on Jennet's plate that morning (and also in the video) was, naturally enough, a room full of balloons. (But Jonathan and company did the cleanup.)

Five of our students received their Ph.D.s between the printing of the last ***conduit!*** and the time this issue went to bed. They are: Ted

Camus—“Real-Time Optical Flow;” David Langworthy (passed defense only)—“On the Use of Asynchrony in Achieving Extensibility and High Performance in an Object Storage System;” Jak Kirman—“Predicting Real-Time Planner Performance by Domain Characterization;” Kathryn Sanders—“CHIRON: Planning in an Open-Textured Domain;” Sairam Subramanian—“Parallel and Dynamic Shortest-Path Algorithms for Sparse Graphs.”

The photograph on Page 11 shows four out of the five along with their rubber chickens, the department’s token that a person has successfully defended his or her Ph.D. thesis. The conferring of the rubber chicken (which is thrown at, not handed to, the successful candidate) goes back to 1985 or so when, after a particularly bad invited talk, one graduate student commented to another that someone should have shut the speaker up by throwing a chicken or somesuch at him. It turned out that the person suggesting this was due to defend his Ph.D. thesis a few weeks later, and several of the graduate students thought that during the defense they would throw a chicken, or at least a rubber chicken, at him. They chickened out (so to speak) and threw it only after the talk, but a tradition had started.

van Dam fashion clones: What do yuppies do when the weather gets cold?
They throw another sweater around their shoulders!

conduit!

A publication of The Computer Science Department
Brown University

Suzi Howe . . . . Editor-in-Chief
Katrina Avery . . . . . Writer & Editor
Jeff Coady. . . . Technical Support

Inquiries to: ***conduit!***
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912
FAX: 401-863-7657
PHONE: 401-863-7610
EMAIL: sjh@cs.brown.edu
WWW: http://www.cs.brown.edu/publications/conduit/

Department of Computer Science
Brown University
Box 1910, Providence, RI 02912

conduit!

NON-PROFIT
U.S. Postage
PAID
Providence, RI
Permit #202

Printed on recyled paper

Address changes welcomed