

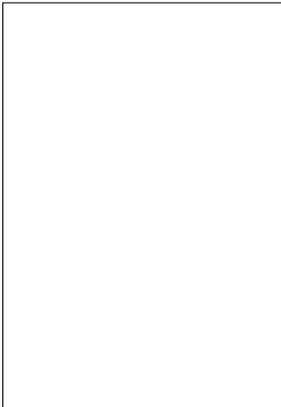
conduit!

Volume 1, Number 2

Department of Computer Science
Brown University

September, 1992

1992 AAAI ROBOT EXHIBITION AND COMPETITION



Thomas L. Dean

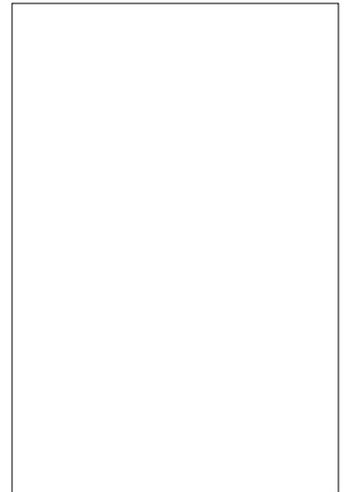
This past June, Computer Science Professor Tom Dean designed and organized the first American Association for Artificial Intelligence (AAAI) Robotics Exhibition and Competition, held in conjunction with AAAI's Tenth National Conference in San Jose, CA. This event featured robots, videos, and images from university and industrial research laboratories around the world. It stressed the range of tasks robots must master to move about an unstructured environment and cope with interference, both deliberate and inadvertent, from the humans with whom they share that environment. These robotics applications focus on apparently mundane tasks that are in fact every bit as complicated as traditional AI's master-level chess in terms of the interplay of behaviors and the physical interaction of the robot with the real world; moreover, they demand a degree of autonomy and robust execution unparalleled in prior commercial applications of AI technology.

Background and Planning

While a number of robot competitions have been run at conferences in the past, most have stressed hardware or low-level control issues not particularly appropriate for an AI conference. It was certainly easy to imagine a competition that stressed more complicated forms of reasoning; the trick was to stage it within the constraints imposed by a large national conference. Dean believed, however, that the research community's hardware and software were robust enough that staging such a public event could bring credit to the field and instruction and entertainment to the general community. The goal was to

design a competition that was feasible within the current state of the art, exercised capabilities of interest to the AI community, could be carried out in the three-day conference timespan, and was entertaining to watch. The set of tasks chosen involved interaction with people, navigation and exploration in an unknown environment, path planning, and command following. The hope was to encourage machines that exercise common-sense capabilities involving planning, control and rudimentary learning and spatial reasoning.

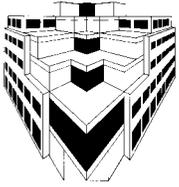
The competition involved three events, one on each day of the conference, and two distinct stages. The first event required the robots to navigate in a cluttered environment and interact with people; this initial stage served as a qualifying round for the subsequent events. In the second event, the robots explored their environment to find and identify ten designated objects. Finally, in the third round, they were to carry



Brown's Huey

out commands issued by the judges to visit specific sets of designated objects in specific orders; here they could use any information acquired in the second event to enhance their performance. The second and third events comprised the second stage of the competition, and prizes were awarded for each of two stages





as well as for a number of special categories suggested by the judges.

The Competitors Arrive

Ten robots arrived in San Jose ready to compete: from Brown University ('Huey'), Carnegie Mellon University ('Odysseus'), Georgia Institute of Technology ('Buzz'), IBM Watson Laboratories ('TJ'), MITRE ('Uncle Bob'), NASA ('Soda Pup'), SRI ('Flakey'), the University of Chicago ('Chip'), and the University of Michigan ('CARMEL'); the tenth entry ('Scarecrow') was built by David Miller and his son, Jacob Milstein. They all had remarkably similar basic architectures, consisting of a multi-level control system coupling low-level routines for sensing, primitive navigation (e.g., wall-following) and obstacle avoidance operating continuously in highly responsive feedback loops, and high-level routines for planning and map generation that relied on the low-level routines to keep the robot in synch with real-time events. There were systems based on variants of Brooks' subsumption architecture (IBM Watson) and on Rosenschein and Kaelbling's situated automata approach (MITRE). Some systems adopted no particular architecture but

"In general, the software developed for this competition was remarkable for its sophistication and ease of modification"

rather incorporated the lessons such architectures have inculcated over the years: for instance, Brown's entry (designed and run completely by undergraduates, unlike all the other entries) was a modular object-oriented software system that

used multiple processors to manage low-level and high-level routines asynchronously. In general, the software developed for this competition was remarkable for its sophistication and ease of modification. In some cases, code was revised extensively during the competition itself to cope with hardware failures and complications introduced by specifics of the conference hall.

Day One

The first day's events revealed a number of problems with robots' remote communications to computing devices via modems—the competition hall was a veritable soup of rf noise, what with the portable microphones, transmitters

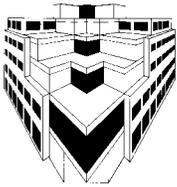
used by the press, two-way radios used by convention center employees, and dimmers and starters for the powerful halogen lighting. They also revealed a great number of extra features provided by various robots' designers: many

"I've got to rest before I fall apart"

robots were extremely agile and even graceful, and some featured computer-generated sounds and voice synthesizers (MITRE's Uncle Bob punctuated its exploits with "I need a vacation" from Arnold Schwarzenegger in Terminator II and "I've got to rest before I fall apart" from Star Wars' C3PO). Miller and Milstein's Scarecrow, while a real crowd-pleaser, did not fare well in this first stage because its primary mode of locomotion involved crashing into objects and obstacles and then dashing off in the opposite direction! The winner of this first stage was TJ, the IBM Watson entry, which combined ultrasonic sensors (sonar) for long-range obstacle detection with near-infrared sensors for short-range obstacle detection to achieve a robust, agile navigation and obstacle-avoidance system.

Day Two

On the second day, the competition rings were cluttered with obstacles constructed from cardboard boxes, as in the first stage; in addition, ten 'objects'—eight-foot poles constructed from PVC drainage pipes—were to be identified. This event was meant to encourage entrants to construct some sort of internal representation of the ring, encoding the location of the objects for use in the contest's final stage in which robots had to visit a set of objects in a particular order. The different teams could rig the PVC poles with whatever sensor stimuli they wished: bar codes, colored rings, reflecting tape, and sonar beacons were all employed by one team or another. The robots were to explore the ring and identify all the objects. Robots that could not identify objects by labels were forced first to differentiate a pole from a box—this is actually pretty tricky using sonar alone—and then to distinguish each object from the others using object location. Robots attempting this strategy, like Brown's entry Huey, were unable to compete with those using the specific stimuli attached to the poles for identification. Huey's software for interpreting the sonar data, while sophisticated, proved very slow, and its performance in this second event was so poor that it was eliminated from the



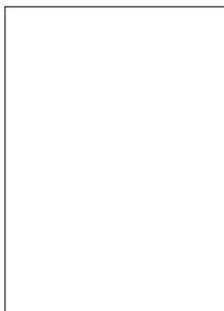
final stage of the competition. Scarecrow continued to perform well and also entertained the audience in this event by its lurching gait; in addition, the effectiveness of its random-walk strategy offered a dramatic illustration of the theoretical result that a short random walk in an undirected graph will with high probability visit every location in the graph.

The Final Heat

Eight robots had completed the first two days' events and were still in good enough shape to continue to the final heat. Here entrants were asked to visit specific locations in the contest ring in a specific order. Michigan's CARMEL was the leading contender going into this event. In a false start, CARMEL was given the wrong command and confused everyone by flawlessly executing a sequence quite different from that specified by the judges. In its second attempt, it seemed to become disoriented and at one point stood by the wall of the ring scanning the crowd for the obstacle it was looking for—as if an Olympic favorite had failed inexplicably in the last seconds of the finals. Here, however, CARMEL recovered: in its final try it executed the sequence flawlessly and turned in the best time for the event.

In designing the competition tasks, the goal had been to encourage approaches relying on techniques borrowed from and extending current research in planning, learning, and spatial reasoning. It turned out that those entries using simple sensors and sophisticated methods for dealing with uncertainty in sensing and movement were too slow to compete. As was apparent from Scarecrow's performance, sophisticated reasoning was not necessary for passable performance in some events. CARMEL's accurate dead reckoning and long-range sensing were significant factors in its success. That the specially designed robots did well was neither a surprise nor a disappointment. The tension between approaches tailored to a particular problem and those striving for generality is extremely healthy for the field, and future competitions—of which it is hoped there will be many—should continue to play off these very different approaches one against another in an effort to learn more about the basic tradeoffs. The competition drew a great deal of interest and enthusiasm among the attendees and especially among the competitors; many people commented that it had made them think about problems in robotics and how AI might contribute further in this area.

LESLIE PACK KAEHLING, NEW FACULTY MEMBER



Leslie
Kaelbling

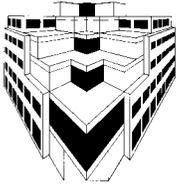
The latest addition to our faculty in the area of artificial intelligence is Leslie Pack Kaelbling. She comes to us from California, having grown up on a farm in Hollister. She completed both undergraduate and graduate studies at Stanford University. While in graduate school she was a researcher at the Artificial Intelligence Center of SRI International, an independent research institution. She then went on to become one of the founders of Teleos Research, a small company doing basic research in artificial intelligence and robotics.

Her main technical interest is in methods for designing and building *embedded agents*: robots that have an extended, complex interaction with a dynamic environment. She worked initially on developing formalisms for programming robots that gave the programmer an easy-to-use declar-

ative language, while compiling into efficient run-time code with real-time performance. After much experience in programming robots to do seemingly simple tasks like delivering objects in an office environment, she found that programming was not enough. The problem is that the programmer rarely knows enough about the robot and the environment to write a correct program. This led her to investigate methods for allowing robots to *learn* how to behave through

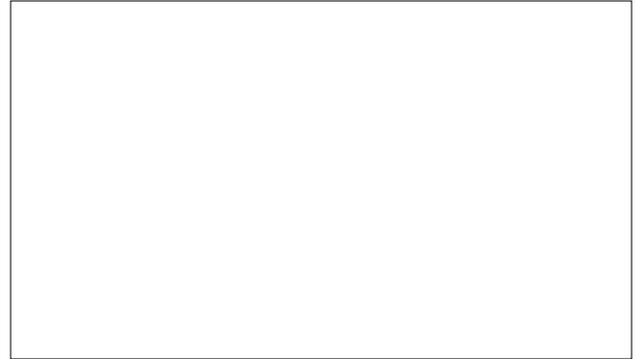
"Her main technical interest is in methods for designing and building embedded agents"

trial-and-error execution of actions in their environments. Leslie's dissertation work was on algorithms for reinforcement learning, and included an implementation of learning on a real robot.



Here at Brown, Leslie is continuing her work in robot learning as well as extending her previous work on programming robots to the area of *active perception*: deciding what actions a robot should take in order to gain information about its environment. She has just received one of the National Science Foundation's prestigious National Young Investigator awards (only 202 of these awards were made this year) with which she will pursue reinforcement learning in robotic agents. She also has a NATO travel grant to collaborate with Professor Tim Smithers at Vrije Universiteit Brussel. They will be using their programming and learning techniques on small robots built by Professor Smithers.

In her free time, Leslie studies learning in another domain—horses. She is an avid rider, even to the point of bringing with her from California her 17.2-hand dark bay thoroughbred gelding, New Moon Bay. He experienced snow for the first time last winter and loved it! She studies dressage, the classical art of horsemanship, and competes at the elementary levels.



Leslie and "Moon" at a dressage show

REASONING VS. MODELING IN COMPUTER SCIENCE

In June 1992 I participated in two significant programming language conferences. The Fifth-Generation Computing Conference in Tokyo examined the achievements of logic programming, while the Simula 25th Anniversary in Oslo celebrated the beginning of object-oriented programming.

The Fifth-Generation project achieved international visibility for logic programming, but was unable to achieve a breakthrough in practical problem solving. This conference marked the end of a ten-year research project that accomplished a great deal but left many unanswered questions. Nationally sponsored Japanese research was shifting its focus from logic programming to "real-world computing." Was the termination of this project after so much investment a strategic error, or is general-purpose logic programming inherently hard or unachievable?

Simula had introduced real-world computing (by object-oriented modeling and simulation) 25 years before the real world was ready for it. The

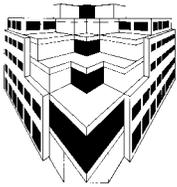
designers of Simula were pleased that the real world was finally catching up, but concerned that current object-oriented languages had sacrificed principles for efficiency and portability. *De facto* standards imposed by

*"Simula introduced
real-world computing
25 years before
the real world was
ready for it"*

language usage were compromising future language development, but there was little alternative because of the slow convergence on common principles and models. Is the elusiveness of tidy object-oriented principles and formal models due to insufficient effort, or is this problem, too, inherently hard or unachievable?

In [We1], we argue that general-purpose logic programming and formal models for object-oriented programming are unachievable for essentially the same reason: namely, limitations in expressiveness of deductive reasoning. Here we examine the historical roots of the dichotomy between logic programming and object-oriented modeling in terms of the philosophical debate between rationalism and empiricism.

Peter Wegner



Rationalists assert that physical phenomena are *necessarily* deducible from logical principles, while empiricists believe that laws of nature can be derived only *contingently*, by observation. Logic programming takes no epistemological position on whether the “real world” is inherently logical or contingent. It makes the weaker claim that acquired knowledge (whether acquired empirically or by a magical rationalist oracle) can be logically represented and manipulated. Logic programming is rationalist in its attempt to reduce problem-solving to deduction, while object-oriented programming is empiricist in organizing knowledge according to the inherent structure of application domains.

Descartes' *cogito ergo sum* succinctly asserts the rationalist credo that thinking is the basis of existence. Descartes' success in reducing geometry to arithmetic spurred more ambitious rationalist efforts to reduce empirical sciences and mathematics to logic. Logic programming is Cartesian in its attempt to reduce computation to logic.

Hume was called an empiricist because he showed that inductive inference and causality could not be proved deductively, thereby demonstrating the limitations of rationalism. Kant, “roused from his dogmatic slumbers” by Hume, wrote the *Critique of Pure Reason* to show the limitations

of pure reason as a tool of thought. The emergence of modern science in the 18th and 19th centuries was motivated by empiricism, but

rationalist revivals continued to have great appeal. Hegel, whose “dialectical logic” extended the reach of “pure reason” beyond its legitimate domain, profoundly influenced both social philosophers like Marx and mathematical philosophers like Russell. George Boole, in calling his logic treatise *The Laws of Thought*, demonstrated that rationalism was very much alive in the mid-19th century.

The influence of Frege, Russell, and Hilbert led to rationalist (formalist) domination of mathematics in the early 20th century. However, Godel incompleteness, Turing non comput-

ability, and quantum-theoretic uncertainty demonstrated theoretical limitations of mathematics, computation, and physics. Godel's incompleteness result (that arithmetic cannot be expressed by logic) demonstrates the impossibility of Hilbert's formalist mathematics program. Turing noncomputability shows that certain functions cannot be computed, while quantum-theoretic uncertainty shows that the “real world” cannot be mechanically modeled. The reaction against formalism is exemplified

“There is no golden rule, silver bullet, or logical inference system that can impose rationalist order on empirical chaos”

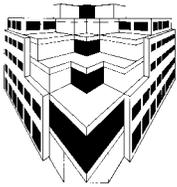
by Von Neumann, who was a disciple of Hilbert in his early years but devoted his later years to computing.

The evolution of artificial intelligence has been dominated by the conflict between rationalism and empiricism. The early rationalist optimism that intelligence could be realized by a *general problem solver* gave way in the 1970s to an empiricist emphasis on domain-dependent knowledge representation. The widespread acceptance of logic as the primary problem-solving mechanism is rationalist, while low-level vision and expert systems are empiricist.

The debate during the early 1970s concerning declarative versus procedural knowledge representation was resolved in a rationalist manner in favor of the predicate calculus. Turing's rationalist hypothesis that intelligence is reducible to (and achievable by) computation dominated the 1960s and 1970s, while distributed and connectionist models staged a comeback in the 1980s. The distributed artificial intelligence view that problem solving is a cooperative activity among distributed agents (as in Minsky's *The Society of Mind*) is empiricist rather than rationalist.

In the programming languages field, the debate between rationalism and empiricism has focused on logic versus object-oriented languages. Early optimism that programming

“Godel's incompleteness result demonstrates the impossibility of formalist mathematics”



languages can describe applications according to rationalist principles is giving way to the realization that reasoning is a restricted form of modeling that cannot easily handle changes in modeling requirements. There is no golden rule, silver bullet, or logical inference system that can impose rationalist order on empirical chaos, even after the knowledge has been acquired. Though some problems have an inherently inferential structure, modeling and simulation of application domains, and especially the evolution and management of such models, can be better handled by object-oriented than logic programming.

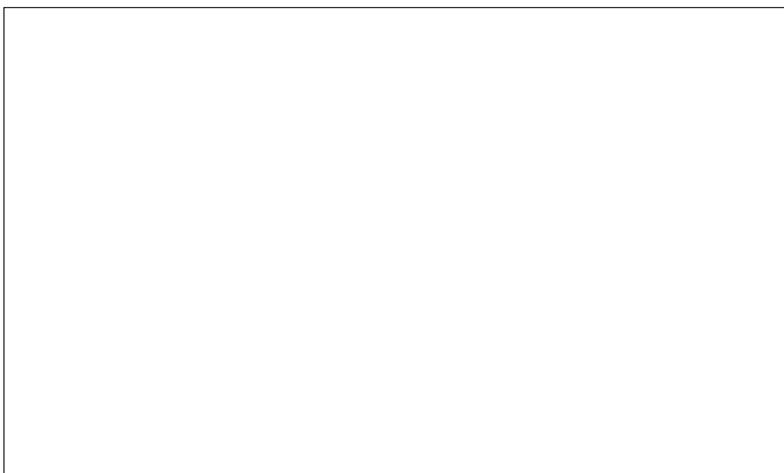
To distinguish between the logic and object-oriented paradigms, a *component-based* model of computation can be developed with *message execution* as the atomic unit of computation, and the metrics that distinguish between the modes of program organization of logic and object-oriented programming can be formulated in terms of modes of component interaction. The concep-

tual differences between problem solving by reasoning and by modeling can be expressed by computational differences between “reductive” and “reactive” computation. Thus computer science can provide an insight into human modes of problem solving by capturing their high-level divide-and-conquer mechanisms for expressing the solution of large problems in terms of interacting subproblems. This description of human problem solving by computational divide-and-conquer mechanisms does not assume the similarity of human and computer problem solving mechanisms, but merely similar high-level mechanisms for managing the complexity of very large problems. Logic programming is inherently more prescriptive and less flexible than object-oriented programming in decomposing very large problems into dynamically interacting autonomous subproblems.

[We1] *Object-Oriented Logic Programming*, Brown University, August 1992.

INDUSTRIAL PARTNERS PROGRAM SYMPOSIUM ON CASE

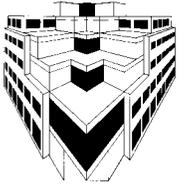
The 10th IPP Symposium was held last March 19. Despite the forecast of a major blizzard on top of the six inches of snow already on the ground, well over half of the 66 registrants attended the day-long event. The topic was



Symposium speakers, l to r: Ken Zadeck, Roy Levin, Dilip Soni, Steve Reiss, Lou Mazzuchelli, John Field

“Software Development and CASE Tools,” with Professor Steve Reiss hosting the program.

Lou Mazzuchelli, VP and Chief Technical Officer of Cadre Technologies, Inc., was the dynamic first speaker. His subject, “CASE: A Reality Check and a Look Into the Future,” generated a great deal of lively interaction. John Field, from IBM’s Software Research Staff, spoke about “Technology for Semantics-Based Programming Tools,” expressing his belief that now we have the computer power, we will see a lot more tools developed using rewrite semantics. Professor Ken Zadeck, with grad student Tony Davis, spoke on “Restructuring Large Programs,” work in which they use graph-based semantic tools. Roy Levin, Senior Consulting Engineer at DEC SRC in Palo Alto, spoke on “Initial Experiences with the Vesta Software Development Environment.” He described a different approach to configuration management; many of these ideas are now being integrated into new systems. Dilip Soni, Senior Research Scientist at Siemens Corporate Research, gave a talk on “Change Assistant,” a program which supports evaluation, planning, and automation of certain classes of changes to large software systems



written in C. Host Steve Reiss, as the last speaker of the day, valiantly managed to squeeze his talk into the 10 minutes that remained! His topic “Program Visualization: Where Do We Go From Here?” detailed efforts to provide a flexible interface to a wide range of program visualizations. The system currently under development at Brown allows the programmer to define visualizations as abstractions using queries over an object-oriented database of information about the program. These abstractions can then be visualized and browsed using easily defined class-based mappings and a generic filtering system.

Demos of the FIELD programming environment, currently being developed at Brown, gave attendees an opportunity to observe close up and to interact with graduate students. Cadre Technologies also gave a demo of the Teamwork environment.



Graduate student Yi-Jing Lin demoing FIELD to IPP visitors

The 11th symposium on, “Progress in Distributed, Object-Oriented Computing,” will be hosted by Professor Peter Wegner and will take place on October first. Besides talks by individuals from member companies and Brown faculty, attendees will give short presentations about current research projects. (See schedule on back page.)

CURRENT RESEARCH OF THE GRAPHICS GROUP

The long-term research goal of the Brown University Graphics Group is to develop easy-to-use, powerful authoring tools for creating interactive illustrations for electronic books. In an electronic book, illustrations are not limited to static images or to prestored sequences that allow only passive viewing. Interactive illustrations are used that are based on simulation and real-time animation; such illustrations should be able to model a

“We will investigate the novel idea of a time-varying user interface”

diverse class of phenomena and allow real-time interaction so the user can better understand the objects and phenomena described.

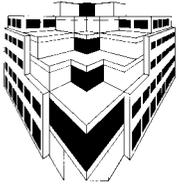
Currently, and for the next several years, our research will focus on time-parameterized



RACK—By using direct manipulation and intuitive feedback, this 3D interface allows the user to bend, twist, and taper objects easily

modeling (sometimes called 4D modeling). We have completed the framework for a large-scale object-oriented modeling and animation system. On top of this framework is FLESH, a powerful scripting language that allows the specification of geometric and visual attributes on objects. Recent extensions to FLESH provide a Lisp-like syntax for specifying complex behavioral attributes on objects.

Another aspect of our current research is the creation of a powerful user interface for the

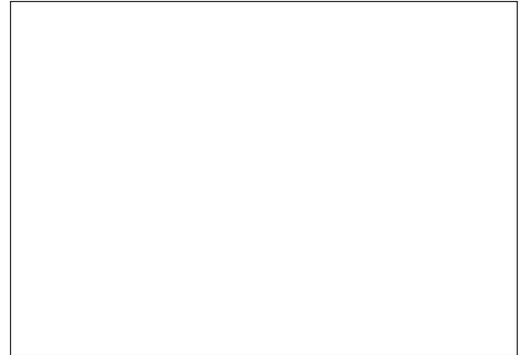


modeler. The components of the user interface are built *into* the system with FLESH instead of on top of an underlying system. Since these components are themselves objects in the system, with time-varying behavior, we will investigate the novel idea of a time-varying user interface. This research is currently focused on 3D widgets that provide intuitive functionality for complex object manipulations that cannot be obtained from normal 2D widgets. The user interface capabilities will play a central role in our research towards the design of a unified framework for handling anything from conventional 2D widgets to 6D input devices.

We have also been studying the unification of two types of modeling, trying to develop conversions between sampled volumetric models and surface-based models. Our intent is to allow the use of surface- and volume-based modeling techniques simultaneously, thus breaking down the Tower of Babel that has grown up in modeling.

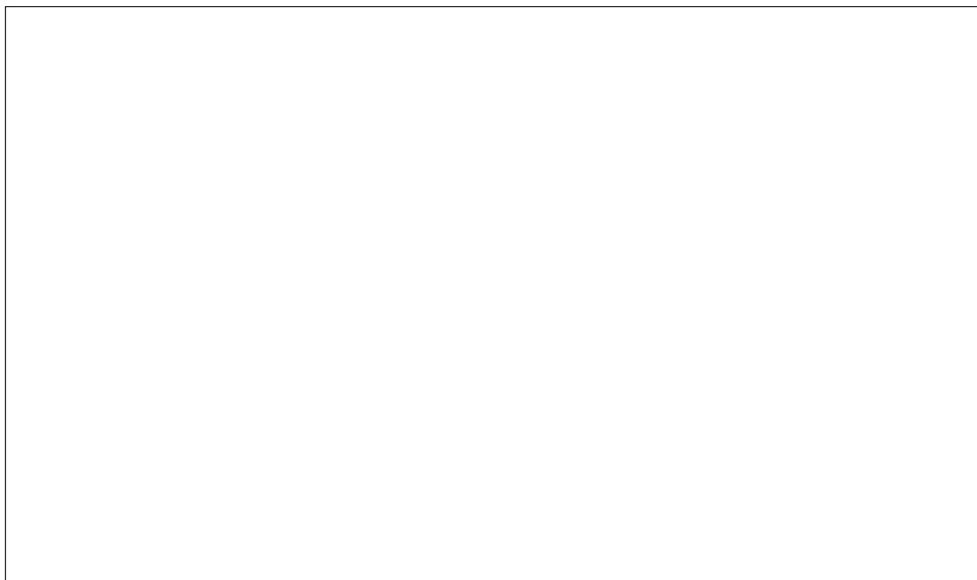
Finally, we are working on finding ways to compromise in a continuously variable way between speed and quality of illustrations, so that as a user pauses (or indicates a willingness to wait), the system can improve the quality of the image (or simulation, or solution to some problem) that is being shown. The intent is to recognize that sometimes it is better to have an approximately correct answer now than a per-

fect answer later. For example, a picture showing the relevant part of a machine in great detail and remote parts as line drawings may in some contexts be nearly as useful as a fully rendered image of the whole machine. As another example, in a simulation based on Newtonian physics, collision detection between geometric objects is a very compute-intensive process. Approximations to an object's extent allow much faster collision detection at the risk of errors. We are working on the automation of such tradeoffs.

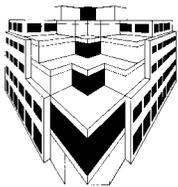


SHADOWS—Shadow projections of modeled objects allow the user to work easily in 3D world by combining many views of an object into one window

(Members of the Graphics Group away when the photograph below was taken are: Philip Hubbard, Cindy Grimm, Brook Conner, Mike Anderson, Chris Brown, Chris Nuzum, Oren Tversky, and, of course, Andy van Dam.)



The Graphics Group, l to r: Bob Zeleznik, David Yang, Andrew Kunz, Noah Parker, Tom Meyer, Dan Robbins, Lori Agresti, Jeremy Katz, Larisa Matejic, John Hughes, Matthias Wloka, Nate Huang, Ken Herndon



John F. Hughes. John attended the ACM SIGGRAPH conference in Chicago where he had three papers; out of the 44 papers given, only one other person had as many as three.



Leslie P. Kaelbling. Leslie has presented three conference papers this year, one of them at the European Conference on Artificial Life in Paris. She was also an invited panel member at the AI Planning Conference, an invited lecturer at the University of Zurich, and an instructor at a two-week summer school on Comparative Approaches to Cognitive Science in Aix-en-Provence, France.



Paris C. Kanellakis. Paris brought out a book last year: *Building an Object-Oriented Database System: The Story of O2* (edited with Francois Bencilhon and Claude Delobel, Morgan Kaufmann, June 1992). He was chair of the 11th ACM Symposium on Principles of Database Systems (San Diego, CA, June, 1992) and co-chair, with Joachim Schmidt, of the 3rd International Workshop on Database Programming Languages, Nafplion, Greece (August, 1991, proceedings published in March, 1992, by Morgan Kaufmann). In addition, he was guest editor of two special journal issues: one of *Theoretical Computer Science* (with Serge Abiteboul) on the 3rd International Conference on Database Theory, December, 1990, Paris, and one of *JCSS* on the 11th ACM Symposium on Principles of Database Systems, June, 1992.



Philip N. Klein. Philip will be spending the 92/93 academic year as a Visiting Scholar at MIT.

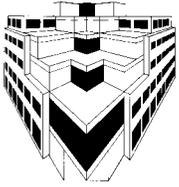
Franco Preparata. Within the framework of his research in parallel computation and computational geometry, Franco was invited to be a visiting professor at the Ecole Normale Supérieure in Paris and at the University of Padova, Italy. He directed two sessions of the Fibonacci Institute in Trento, Italy, of which he was a founder in 1989. He is scheduled to be a keynote speaker at the International Workshop on Parallel Computing in Paderborn, Germany, and at INRIA's 25th anniversary in Paris.



Steven Reiss. Steve presented three conference papers this year, attended the Dagstuhl Workshop on Programming Environments in Germany, and gave invited talks at Silicon Graphics, Bell Labs, and IBM. He was the keynote speaker at IFIP '92.



John Savage. Recently returned from a year's sabbatical at the University of Warwick, England, John has been chosen as an alumni representative on the MIT Corporation Visiting Committee for the Electrical Engineering and Computer Science Department for the period October 1991 through June 1995. While on sabbatical he gave invited lectures at the University of Warwick, Christian-Albrecht University, University College of Wales, and the University of Wales, and presented papers at the British Colloquium on Theoretical Computer Science and at the German Conference and Research Center for Computer Science at Schloss Dagstuhl, Germany. He was editor, with Thomas Knight, of the book *Advanced Research in VLSI and Parallel Systems* (MIT Press, 1992). Earlier this year John was made a Fellow of the IEEE. Because he was abroad at that time, the award will be presented at the IEEE Supercomputing Conference in November.



Roberto Tamassia. Besides presenting six conference papers this year, Roberto also co-organized the International Work Meeting on Graph Drawing, Marino, Italy, June 3-5, 1992 and gave invited lectures at the Fujitsu Laboratories in Numazu and Tokyo, Japan, the University of Rome, Johns Hopkins, and the University of California, Berkeley.

Andries van Dam. In recognition of his major contributions to the field of computer science, our department, and Brown University, Andy has recently been named an L. Herbert Ballou University Professor. He was the plenary speaker at the Hawaii International Conference on Systems Sciences (*Escaping Flatland*) and the keynote speaker at both the Symposium on Interactive 3D Graphics and the Hewlett Packard Peripherals Developers Conference. Andy has been on the Committee to Assess the Scope and Direction of Computer Science and Technology which produced what we expect will be an influential report. This has come out as a book, *Computing The Future: A Broader Agenda for Computer Science and Engineering*, which will be highlighted in the next issue of *conduit!*

Pascal Van Hentenryck. In addition to three conference papers, two book chapters, and two invited talks at international conferences, Pascal is editing a special issue of the *Journal of Logic Programming* on Constraint Logic Programming. The international conferences at which he spoke were COMP EURO 92 (International Conference on Com-

puter Systems and Software Engineering) in the Hague and LPAR 92 (International Conference on Logic Programming and Automated Reasoning) in St. Petersburg.

Jeffrey S. Vitter. Earlier this year, Jeff served on conference program committees for the 1992 IEEE Data Compression Conference and the MIT-Brown Conference on Parallelism and VLSI. He is completing his first year as Vice-Chair of the SIGACT special-interest group of the ACM. Jeff has had six conference publications so far this year, with more on the way. He has given invited talks at several universities, including INRIA in southern France and the University of Victoria.

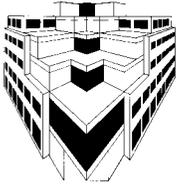
Peter Wegner. This June, Peter was an invited speaker at the Fifth-Generation Computing Conference in Tokyo and at the Simula 25th anniversary meeting in Oslo. He is co-editor of *Object-Based Concurrent Computing*, recently published in Springer Verlag's series of Lecture Notes in Computer Science. His paper "Dimensions of Object-Oriented Modeling" will appear in a special issue of *IEEE Computer* in October, and his paper "Megaprogramming" will appear in the *Communications of the ACM* in November. He will chair a panel on megaprogramming at OOPSLA in October. As Editor-in-Chief of ACM Press Books, he has been responsible for the more than 20 books published in the last three years under the imprint of the ACM (jointly with Addison-Wesley).

FROM THE CHAIRMAN, Eugene Charniak

Several things have happened at the Department since the last issue of *conduit!* First and foremost, we are happy to welcome Robert (Rob) Netzer as a new faculty member. Rob has just received his Ph.D. from the University of Wisconsin. His area of research is parallel and distributed debugging. (For

example, one problem with debugging parallel programs is that the order in which things take place can differ from run to run and the bugs might only surface when things are done just so. Some of Rob's work makes it simpler to detect such situations and fix them when they occur.) We all liked Rob's work because it was a nice blend of theoretical ideas and practical systems building, and we are glad he chose to join us here at Brown.

Another news item is that Leslie Kaelbling



received an NSF NYI (National Young Investigator) award. Devoted readers of this column will remember that last time we welcomed Leslie to Brown and promised an article about her in this issue (a promise kept). The NYI awards are given only to the top young researchers and we are very proud that Leslie has received one of these awards, especially so early in her career.

You may also remember that last time I noted that the department was in the process of selecting new workstations to replace our "aging" Sparcstation 1s. I am happy to announce that we have decided to replace them with new Sparc 10s from Sun. It was a hard decision, with both IBM and SUN coming in with very tempting proposals. Probably the most noticeable change will be in our workstation classroom, where we will now be installing Sparc 10s with 3D graphics accelerators.

We believe that 3D graphic capabilities will prove useful in teaching Computer Science courses, and we are happy to acknowledge an NSF grant which helped us to equip the lab in this way. Both Andy van Dam, who teaches our introductory programming course for majors (CS11) and Steve Reiss, whose programming environments are used in several of our courses, are now working on using the 3D graphic capabilities for more effective teaching and debugging.

We are very pleased that, since June of 1991, the following graduate students have successfully defended their doctoral dissertations:

Ajit K. Agrawal: "Network Design and Network Cut Dualities: Approximation Algorithms and Applications." He has a position with DEC's Massively Parallel Systems Group in Maynard, Massachusetts.

Kenneth J. Basye: "A Framework for Map Construction." Ken is now an Assistant Professor at Clark University.

Robert Cohen: "Combine and Conquer." Bob is now a Visiting Assistant Professor at

the Coast Guard Academy in New London, Connecticut.

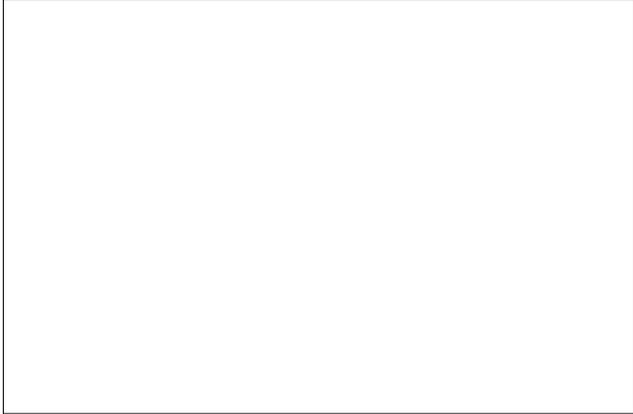
Keiji Kanazawa: "Reasoning About Time and Probability." Keiji is now a research scientist at the University of British Columbia.

Jyh-Han Lin: "Approximation Algorithms and Complexity Results for Machine Learning." Jyh-Han is a member of the research staff at Motorola in Florida.

Mark H. Nodine: "Minimizing the Input/Output Bottleneck." Mark is now a research staff member at Motorola in Cambridge.

Eugene Santos: "A Linear Constraint Satisfaction Approach for Abductive Reasoning." Gene has recently been offered a position at the Air Force Institute of Technology.

Solomon Shimony: "A Probabilistic Framework for Explanation." He is now a

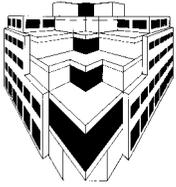


Incoming graduate students relaxing in the 4th floor atrium

Lecturer at Ben-Gurion University. (This roughly equates to an Assistant Professorship in the U.S.)

Alexander Shvartsman: "Fault-Tolerant and Efficient Parallel Computation." Alex is now working as a Technical Leader for DEC in Littleton, Massachusetts.

Incoming graduate students include eight Ph.D. candidates and 16 Master's candidates. Orientation week is currently in full swing after an ice-cream kick-off party on the fifth floor deck.



The 11th IPP Symposium
 Progress in Distributed,
 Object-Oriented Computing
October 1, 1992

- 8:30 Registration and Continental Breakfast
4th Floor, CIT Building
- 9:00 Introductions by Eugene Charniak, Chairman
- 9:15 **Object-Oriented Programming in
the Large**
Peter Wegner, Brown University
- 9:45 **Spring: A Distributed, Object-Oriented
Software Platform**
Steven Gadol, Sun Microsystems
- 10:30 B R E A K
- 11:00 **Concert: A Portable, Distributed Software
Platform**
Shaula Yemini, IBM Yorktown
- 11:45 **Language and Operating Environment
Support for Multi-Applications**
Robert Strom, IBM Yorktown
- 12:30 B U F F E T L U N C H
- 1:30 **Types in Distributed, Object-Oriented
Systems**
Andrew Black, DEC Research
- 2:15 **Research Directions in Distributed, Object-
Oriented Databases**
Stanley Zdonik, Brown University
- 2:45 B R E A K
- 3:15 **Research Directions in Object-Oriented
Environments**
Steven Reiss, Brown University
- 3:45 **Short presentations of current research
projects by attendees**
- 4:45 Wrap-up Discussion
- 5:15 R E C E P T I O N (5th Floor)

© Steve Rosenthal, Cambridge Seven Associates

*The Thomas J. Watson, Sr. Center for
Information Technology. Computer
Science is on the 4th and 5th floors*

conduit!

A publication of The Computer Science
Department
Brown University

Suzi Howe Editor-in-Chief
 Katrina Avery Writer & Editor
 Jeff Coady Technical Support
 Dan Robbins Design Assistant

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

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



conduit!

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