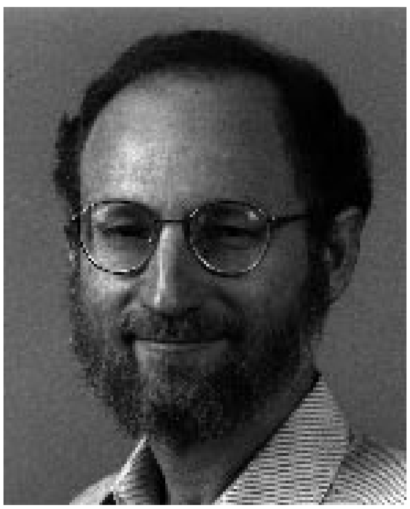
conduit!

Volume 8, Number 2 | Department of Computer Science, Brown University | Fall, 1999

RANDOMNESS, PROBABILITY and COMPUTING



Eli Upfal

"God does not play dice with the universe" was Albert Einstein's famous objection to modern quantum-mechanics theory. This and similar objections reflect the human conceptual difficulty in grasping a world that is not governed by fully deterministic laws. Nevertheless, research in the past century has demonstrated the dominance of random behavior and statistical laws in almost any field of science, ranging from sub-particle physics to free market economics. Computer science is no exception: from the highly theoretical notion of probabilistic theorem proving to the very practical design of PC Ethernet cards, randomness and probabilistic methods play a dominant role.

Roughly speaking, randomness comes up in two aspects of the study of algorithms: randomized algorithms and probabilistic analysis of algorithms. Randomized algorithms are algorithms that make random choices during their execution. In practice, a randomized program uses values generated by a random number generator to decide the next step at several branches of its execution. For example, the Ethernet access protocol, implemented in the Ethernet card, uses random numbers to decide the exact timing by which it tries to access the Ethernetcommunication medium. Other commonly used applications of randomized algorithms are Monte-Carlo simulations and primality testing for crypto-graphy applications. In these and many other important applications, randomized algorithms are significantly more efficient than the best known deter-ministic solutions. Furthermore, in most cases the randomized algorithms are also simpler and easier to program.

A randomized program can be viewed as a conventional (deterministic) program that reads two streams of inputs: one is the actual input of the program, the other is a sequence of randomly generated numbers. This view raises an intriguing philosophical question: how can a string of random numbers (perfect noise) that adds no information to the program actually improve the performance of its execution? Theoretical computer science still does not have a full explanation of that phenomenon, though more and more applications take advantage of it.

> *"how can a string of random numbers (perfect noise) that adds no information to the program actually improve the performance of its execution?"*

Another algorithm-related application of probability theory is in analyzing the average-case performance of algorithms. Algorithm and complexity theory tries to classify computation problems according to their computational complexity, in particular distinguishing between easy and hard to solve problems. For example, complexity theory shows that the famous traveling salesman problem is NP-hard. Thus it is very unlikely that it has a solution significantly faster than enumerating all possible routes—a task that is exponential in the number of cities and thus not feasible for a large number of cities.

An embarrassing phenomenon for the classical ‘worst-case’ complexity theory is that problems classified as hard to compute by the theory are often easy to solve in practice. Probabilistic analysis gives a theoretical explanation of that phenomenon, namely that these problems are hard to solve on some small set of pathological hard inputs but are actually easy to solve on most inputs, particularly those that come up in real-life applications—scheduling, the traveling salesman problem, packing and covering are just a few examples.

Dynamic Algorithms

The focus of my research is the application of probability theory in computer science. In particular, I am currently interested in studying dynamic algorithms through stochastic an-‘alysis. Research in theoretical computer science has focused mainly on static computation problems, where the input is known before the start of the computation and the goal is to minimize the number of steps till termination with a correct output. However, many important processes in today’s computing are dynamic or interactive processes, whereby input is continuously injected to the system and the algorithm is measured by its long-term steady-state performance. Examples of dynamic processes include communication protocols, memory-management tools, and time-sharing policies. The goal is to develop new tools for designing and analyzing the performance of dynamic processes, in particular through modeling the dynamic process as an infinite stochastic process.

A key feature of dynamic algorithms is the need to make decisions ‘on-line’ without information about future requests. An obvious example is cache maintenance protocols. A cache is a fast memory that serves as a buffer between the processor and a slower but larger memory module. Since a cache is typically small, the program often requests pages that are not in the cache. When a new page is brought to the cache, some other page must be returned to the main memory. Accessing that page later will be expensive, and thus the algorithm tries to evict from the cache pages that are less likely to be requested in the future. The algorithm makes these decisions, of course, without actually knowing the future sequence of requests.

As in the case of static algorithms, randomness is introduced into dynamic computation through the algorithm, the input or both. Many interesting dynamic protocols, such as the Ethernet access protocol mentioned before, are random. An execution of a dynamic random algorithm, even on a fixed input sequence, defines an infinite stochastic process in which a state at a given step depends on the history of the process. Analysis of such a process requires a different approach and different tools than those used in analyzing the finite execution of a randomized static computation.

Stochastic Analysis of Dynamic Algorithms

Worst-case analysis rarely gives interesting insight into the actual performance of a dynamic algorithm. A worst-case adversary can generate extremely hard sequences of requests, and the performance of the algorithm on these pathological cases does not accurately

“Worst-case analysis rarely gives interesting insight into the actual performance of a dynamic algorithm”

represent its efficiency. To offset the effect of rare cases, it is useful to analyze the performance of dynamic systems under some stochastic assumptions on the stream of inputs. Such assumptions are more realistic in dynamic settings, especially when requests are originated by a number of independent processors, than in static analysis. The stochastic process that controls the stream of requests might be stationary, periodic, or even bursty. The goal is to obtain results that are valid under the weakest set of assumptions. The advantages and practicality of this approach have been well demonstrated by the achievements of queuing theory. Our goal is to apply similar techniques to dynamic computer processes that do not fit the queuing theory settings.

Stochastic analysis of dynamic processes builds on the rich theory of stochastic processes, in particular queuing theory, and the theory of stationary processes. However, in many cases new tools are needed to address the specific problems posed by computer-related processes that are discrete and involve complicated dependency conditions.

Recent Work

PhD candidate Gopal Pandurangan and I are studying an improved protocol for admission control in fast communication networks. Modern communication protocols such as ATM (asynchronous transfer mode) achieve high utilization of channel bandwidth by multiplexing communication streams with different flow characteristics into one communication channel. Requests for communication are submitted to the network management protocol with some statistical characterization of the required communication. The network (flow) management protocol uses this information to *statistically* multiplex as many communication requests as possible while maintaining global network performance. Next-generation communication networks are expected to provide QoS (quality of service) guarantees when satisfying communication requests. In particular, QoS protocol is expected to limit to a pre-specified value the probability of communication failure

Gopal Pandurangan

due to events such as link and buffer overflow, packet loss, jitter, etc.

Our new protocol employs an efficient Monte-Carlo method for estimating the failure probability of a network. The tight estimate allows the protocol to utilize the network bandwidth fully without violating the QoS requirements. The new method is particularly useful in a dynamic setting in which communication requests are dynamically added to and eliminated from the system. The amortized cost in our solution of updating the estimate after each change is proportional to the fraction of links involved in the change rather than to the total number of links in the network. We are currently running an extensive simulation project to evaluate the performance of the new technique under various network and load settings.

Course

The research area combines two disciplines, algorithm theory and probability theory. My new course 'CS155: Probabilistic Methods in Computer Science' exposes students to the interplay between these two areas. They study basic probability theory, in particular discrete probability theory (which is more relevant to computer science applications) and then apply this theory to design and analysis of randomized computer algorithms for a variety of applications.

SAVAGE HONORED ON HIS 60th

On September 21, in honor of John Savage's 60th birthday, Franco Preparata hosted a one-day technical forum in the department on "Algorithmic Research." The event recognized, in its subject and character, John's constant devotion to productive and well-regarded academic research. Colleagues and former doctoral students dedicated to the honoree some of their more recent work. The department participated actively in the event, recognizing John's many contributions as a co-founder of our department and his assiduous stewardship as chairman and excellent citizen. After initial remarks by Tom Dean and Andy van Dam, Franco kicked off the day with a talk on "Reconstructing a sequence from its samples: DNA sequencing at the information-theory bound," Charles Fiduccia (PhD '73, Division of Engineering) spoke on "Optimal monotonic search," Eli Upfal on "Reducing network congestion through balanced allocation," David Carlson (PhD '80) on "New insights into two old algorithms: the Fast Fourier Transform and metropolis Monte-Carlo method," and Roberto Tamassia on "Graph drawing and information visualization." The intense technical program had a relaxing break during the buffet luncheon and was capped by a gracious departmental reception (with birthday cake, of course).

Back, l to r: David Carlson, Roberto Tamassia, Eli Upfal. Front, l to r: Charles Fiduccia, John Savage, Franco Preparata

INSIDE IJCAI '99

Tom Dean in Stockholm

Nearly four years ago, at the 1995 IJCAI (International Joint Conference on Artificial Intelligence) conference in Montreal, I was asked to be program chair for the 1999 IJCAI in Stockholm and also to serve on the board of trustees for IJCAI Inc., the governing board that runs the IJCAI conferences, oversees a trust fund, determines conference venues, awards prizes, and generally plays an important role in the field through the various activities associated with the biennial IJCAI conferences.

IJCAI is the premier international conference in AI and the invitation from then-president Ray Perrault was very flattering. Stockholm was one of my favorite cities, I enjoyed working with Ray and the other trustees, and I thought that my experience running other conferences (in addition to smaller conferences and workshops, I was program chair for NCAI in 1991—the National Conference on AI, often called AAAI for the North American professional association that runs it) would make this new task relatively straightforward. I naively agreed to serve.

I can't begin to say how unprepared I was for the experience. NCAI is managed by a well-run professional organization with relatively deep pockets. Much of what was necessary in running AAAI-91 was handled without my being aware of it by Carol Hamilton, the Executive Director for AAAI, and her excellent staff. I was oblivious of many of the details of running a large conference. Even though IJCAI is a larger conference than NCAI and much more complicated in its international focus, most IJCAI work is done by the trustees and numerous volunteers from the academic and research communities. The trustees turn over regularly, so corporate memory is short. For the most part, each IJCAI is invented anew, with little help from past experience.

Each IJCAI has a program chair in charge of the technical program, a general chair in charge of the whole process, a conference arrangements chair, and numerous other local and special program chairs. I worked most closely with Prof. Luigia Carlucci (Gigina) Aiello, the general chair and a professor at Università di Roma "La Sapienza," and Prof. Anita Kollerbaur, the conference arrangements chair and a professor at Stockholm University and Royal Institute of Technology. Gigina and Anita were wonderful to work with, but our job was incredibly frustrating and a bit unnerving as we tried to figure out how to run a conference with a more than million-dollar budget (and no small amount of financial risk) starting pretty much from first principles.

"Running an international conference is fraught with all sorts of politics. You have to make sure that all the appropriate nationalities are represented"

Well, we didn't really start from first principles; Anita knew a lot about planning large projects (part of her academic expertise) and she drew upon the skills of the administrative and technical staff at her university and hired a professional conference organizer to help with various aspects of the planning and execution. I arranged for Carol Hamilton and AAAI to help with the extensive correspondence and paper-handling associated with the technical program. In fact, we had to put together a small army of people to help with the conference, and Gigina, Anita and I had to direct the troops.

Running an international conference is fraught with all sorts of politics. You have to make sure that all the appropriate nationalities and professional societies are represented. Even though the conference was to be held in Stockholm, it was supposed to be hosted by all the Scandinavian countries. Registration typically doesn't cover the cost of running a conference, and so to break even we needed to get sponsors willing to help out financially but not require their corporate logos to be tattooed on the foreheads of the attendees.

Much of my effort was directed at putting together the program committee, finding chairs for workshops and tutorials, and choosing invited speakers and cajoling them to attend even though we could only cover a portion of their overall expenses. I enlisted the help of 37 top researchers to serve on the program committee, and these folks in turn enlisted the help of nearly 400 reviewers to handle the almost 800 submissions and over two thousand reviews. I won't bore you with the details of coordinating this effort and coping with inevitable glitches and embarrassments that occur when dealing with so many people and trying to adhere to a strict deadline determined by printers, international shipping rules, and of course the actual date of the conference at which the final proceedings are to be handed out to attendees.

By the end of last February, we had selected 195 papers for presentation, 14 invited speakers were lined up, 29 workshops and 20 tutorials were on the program, and we had made the selections for the Computers & Thought and Research Excellence awards. The locations for the conference, the receptions, the banquet and other events were settled. Given the chaos of reinventing IJCAI and the fact that none of us were exactly sure who was responsible for what, I was relieved that Gigina, Anita and I were still speaking with one another. After what seemed like two years of steady work, I thought my job was over.

Unfortunately, there were some little details yet to go: putting together the proceedings, developing the content for the brochure, providing a schedule for the conference program, creating paper sessions and assigning them to rooms, finding reliable session chairs and people to introduce invited speakers (92 people), and a myriad of other details. It seemed that I would never get out from under the yoke of IJCAI.

My wife Jo and I arrived in Stockholm on July 30, several days before the official beginning of the conference. There were tutorials and workshops to check on, a robotic soccer competition that was co-located with IJCAI, and trustees' meetings to attend. Indeed, trustees' meetings continued throughout the conference, dealing with the previous conference (in Nagoya), the current conference, the next conference (which is to be held in Seattle and for which the planning was well under way), deciding on the location for 2003 (Mexico), and dealing with various aspects of planning for 2005, 2007 and beyond.

Skeppsholmen

The weather was perfect. Stockholm is a beautiful city and it is great fun to tour its shops and museums, flit from island to island on its convenient water taxis, take day trips out to the archipelago, and relax in its ample parks and gardens. With the exception of a few rain showers, the perfect weather, dry and warm but not hot, continued throughout. The conference attendees were treated to a boat trip and a banquet on a fortress island. It stayed light till 10 pm or so and most people took advantage of the light to wander in the old town (Gamla Stan), enjoy a walk around Skeppsholmen, or take in the rides and other amusements in Djurgarden. The trustees plotted and planned and were rewarded for their efforts by a wonderful dinner in a restaurant located in one of the royal game parks. There were more courses than I could count and fabulous wines that I'll not likely sample again any time soon. One of the trustees knew a little about wine and the steward took us to visit their cellar; along one wall was a collection that included one bottle from each of the major houses in the Bordeaux region, starting from 1900 and extending to the present. The only missing years were those in which war had eliminated an entire year's production.

I heard very few technical papers at IJCAI '99 and, though I heard parts of almost every invited talk, for one reason or another I was never able to sit through an entire talk. After the last of the technical sessions, I felt completely drained. Luckily several of Jo's family had arrived in Stockholm the previous day and so I had plenty to distract me from my post-partum blues. Jo's sister Nancy and her daughter Lila had flown in from Paris, Nancy's husband had flown in from Boston, and Lila's boyfriend came from Rome right after he finished work on Friday. We spent a pleasant weekend enjoying what the conference's attendees had been experiencing the week before.

In retrospect, it was a grueling four years. The final result was very gratifying but I don't know that I would do it again. Delayed gratification is common in the academic world: you submit a paper or proposal and wait months for a response. At this point in my life, however, I think I want my feedback a little more immediately. Running a conference like IJCAI is a bit like building a house: you really can't enjoy the house until the roof is on, the plumbing working, the walls painted, and it's ready to move in; in the interim, there is always something to do, most of it tedious. Still, I have to admit that it's pretty exciting when it all comes together. Today someone called me about running an international conference on ...

Peter and Judith Wegner at the Rehabilitation Hospital of RI in N. Smithfield, where Peter had been undergoing physical therapy. Happily, he is now back at home on the East Side

PETER WEGNER ON THE MEND

The following article and photograph above are by senior news writer Kristin Cole. They appeared in the October 15-21 issue of the ***George Street Journal****.*

The music of Brahms filled the London hospital room where Peter Wegner, a retired Brown computer science professor, had only recently regained consciousness after being in a coma for four weeks.

The bus that struck him as he walked toward Trafalgar Square a month earlier had smashed an elbow, broken ribs, and resulted in a head injury from which doctors had initially predicted only a 5- to 10-percent chance of survival.

When asked whether he liked Brahms, Wegner replied, "Yes, especially Boston Brahms" —a pun referring to the upper-class intellectual Brahmins of 19th-century Boston. For friends and colleagues who had given Wegner a chair engraved "to an unrepentant punster" as a retirement gift, it was perhaps the surest sign that he was on the road to recovery. "It was reassuring to know it was the same old Peter," said Judith Wegner, his wife of 43 years. "He beat the odds."

Wegner is now at the Rehabilitation Hospital of Rhode Island in North Smithfield. He returned to the United States Sept. 30, nearly four months after the accident. On June 11, Wegner had been planning to take a walk in Trafalgar Square, one of his favorite haunts. He had just retired after 30 years of teaching at Brown and was looking forward to a summer filled with engagements.

His first stop in England was to attend a reunion of Jews who, as children, had escaped Nazi terror in their homelands when they were accepted by England as political refugees. Wegner had left Vienna as a 6-year-old, one of 10,000 children on the "*Kindertransport*." After the reunion, he was slated to deliver speeches to professional conferences in Portugal and Scotland. To cap the European trip, Wegner was to receive Austria's highest academic award, for his lifetime contribution to the field of computer science.

His wife had left Wegner only a short time before he headed to Trafalgar Square, returning to her sister's home, where they were staying, to help prepare dinner. But when the meal was ready and Wegner had not arrived home, the family began to worry. They notified police and learned a short while later that Wegner had been taken by helicopter from the accident to the Royal London Hospital. His wallet had not contained information for contacting his sister-in-law. But the wallet *did* contain names of colleagues, including Brown faculty members Andries van Dam and Thomas Doeppner.

Initial reports about Wegner's condition were pessimistic, and the couple's sons, Mark, Jeremy and Michael, flew from the United States to London to be at the hospital. In the weeks that followed, the family updated friends and colleagues on Wegner's status by e-mail, maintaining a list that reached across the Atlantic, including many at Brown. Additionally, Judith notified her husband's professional commitments of the situation.

In the hospital room, Judith and her sister, Marion Rosenberg, sang familiar folk songs to Wegner in English and German, in which he is fluent. They celebrated his 67th birthday there, papering the walls with cards that wished him well and often contained puns. When Wegner first came out of the coma, a tube in his throat prevented him from speaking, but it was obvious he could understand what was being said and recognized his family, said Judith. "That is an unusually long coma for someone to recover from. He is a fighter. He is a determined person and he is working very hard at it now."

In therapy Wegner has strengthened his ability to walk, read and write. A recent milestone in his recovery was walking around the perimeter of the hospital building without the aid of a cane. "Judith has helped me enormously during this time," he said. Wegner looks forward to the day he will return to his office; although retired, he still plans to continue his research. There are also retirement dreams to satisfy, said Judith, such as cruises to Alaska and the Greek Islands. And he has yet to receive the Austrian Medal of Honor for Science and Art, which the government offered to confer in the hospital. Wegner refused. He wants to travel to the country to receive the award, as planned.

They are all plans that friends and colleagues thought Wegner would someday be able to satisfy after learning that his ability to make puns was intact only days after coming out of the coma. "It certainly showed the sign that he was recovering," said Doeppner, associate professor of computer science. "He can come up with the most amazing puns in zero time." Judith maintains a list of Wegner's recent puns at their home in Providence, where the chair with the engraving "to an unrepentant punster, indefatigable scholar, and generous friend" awaits his return.

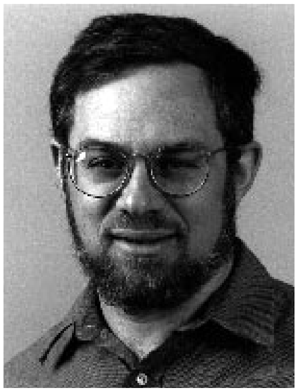
Said Ed Lazowska in his Andyfest kick-off speech, "It's hard to believe Andy is 60. He looks only 45, same as he did when he was 30!"

Scenes from last May's "Andyfest," a two-day symposium in celebration of

(Not ZZ Top) Andy and, l to r, Hendrik-Jan Thomassen, Steve Carmody, Wolfgang Millbrandt and Ken Sloan

Guests seated for lunch on campus

VISUALIZING PROGRAM EXECUTION



Steven Reiss

Introduction

Today's software systems are complex, and their behavior, i.e. what happens as they execute, is typically even more so. Here one must deal with the interaction of large amounts of code, distributed computations, external systems, multiple threads of control, locks, asynchronous events, message traffic, and other complicating issues. However complex this behavior may be, programmers must be able to understand what is going on, especially when the systems start to act in an unexpected or erroneous manner.

Our approach to this problem of software understanding is to use information gathered as the system executes to give the programmer sophisticated visualizations describing the execution. The amount of data that can be collected is vast and can easily be overwhelming. Visualization uses the brain's sophisticated recognition abilities to quickly find relevant patterns in a sea of data in order to make execution understanding practical.

Our research here involves four related but separable problems. The first involves obtaining and storing the data. The second involves letting the programmer specify what needs to be visualized to understand a particular aspect of the behavior of a particular system. The third involves processing the raw trace data so as to obtain the information needed for the visualization. The final problem involves actually producing a visualization with which the programmer can interact to obtain the desired insights.

Obtaining the Data

A lot of data is needed to describe the execution of a system. This includes function calls and returns; thread creation, destruction, and state changes; the state of locks and other synchronization mechanisms; memory access and paging behavior; local control flow; timing information, both real-time and execution-time; message passing; files loaded and used; and file and socket utilization. The general rule for software understanding is that you want to collect as much data as possible since there is some potential problem for which that data will be the key to understanding.

We currently have two separate systems that can gather much of this trace data. The first, AARD, works with C or C++ programs. It consists of a package Wolf that takes executable files and rewrites them with instrumentation code, and a

Andy's 60th birthday, entitled "The Computer, the Academy, and the World."

Enjoying an outdoor break in front of the new MacMillan building where most of the talks took place

President Gee presenting a handsome photograph of the University at the beginning of the banquet at the Biltmore package Vark that uses the raw instrumentation data to provide data for visualization. The second system, TMON, uses Java's JVMPI interface to collect trace information on Java programs. It handles multiple threads of control without adding any additional run time synchronization and records everything that JVMPI offers.

Both of these systems generate large amounts of data. Although it will vary with different applications, our current best guess is that they typically produce about 1G of raw trace data for every 10 seconds of raw CPU time. (This is for relatively high-level information such as method calls; with low-level information such as basic block entries or memory accesses, this goes up by at least an order of magnitude.) Our research here involves managing this trace data so that we can deal with large, long-running systems.

We take two different approaches here. The first involves using the increase in computer capabilities to provide a hardware-based solution. We are looking at doing traces on machines that have several gigabytes of memory, multiple processors, and high-speed connections to large amounts of RAID-based disk storage. We hope that the combination of such massive workstations with sophisticated trace collection code will let us collect complete traces with minimal impact on the running program.

The second approach involves understanding what information is actually needed for a particular visualization and doing the appropriate analysis and processing as the raw data is generated so that only the processed data is stored. One difficulty here is that we want to let the programmer define what should be visualized and hence have to compute the necessary filters and processors dynamically. A second difficulty is that the processing can be quite sophisticated, involving tracking execution histories, maintaining the execution state, and looking at the operations of multiple threads without adding additional synchronization that would change how the program is behaving.

Determining What to Visualize

One conclusion we drew from our past work on software visualization was that no single or small set of fixed visualizations will address the wide range of questions asked as a programmer attempts to understand the behavior of a complex system. Rather than attempting to define such a fixed set and having it not be particularly useful, we have focused on letting the programmer define what should be visualized and how it should be displayed. By making the foundation of such a system rich enough, we hope to address a wide range of understanding problems and thus make our system practical.

Over 300 participants attended Andyfest at venues both on- and off-campus

One of Andy's first doctoral candidates, Ingrid Carlbom, lands a good one at the roast after the banquet

With Ed Lazowska '72 and John Hughes, who orchestrated the entire event

To this end, we created, as part of our Desert environment, the CACTI front end for defining visualizations. CACTI lets the programmer define the data to visualize by graphically defining sets of objects using fields that are the various domains composing the raw data. It is, in effect, a universal-relation-assumption-based visual query language that can access both databases describing the static structure and symbols of the system and the various dynamic analyses that are available from the processed trace data. It is relatively easy to use, as we have demonstrated by addressing a wide range of specific understanding problems including looking for compiler-generated temporaries, abstracting the dynamic call graph, exploring memory behavior over time, looking for patterns in the call stack over time, and finding methods and variables that are never called.

> *"Raw trace data by itself is not very useful. Not only is there much too much information to create meaningful displays, but the information relevant to a particular problem is typically well hidden"*

Our continuing research in this area will attempt to make the CACTI interface more intuitive, provide a reasonable front end to the large number of potential trace analyses, and handle multiple databases with large numbers of potentially overlapping domains.

Processing the Raw Data

Raw trace data by itself is not very useful. Not only is there much too much information to create meaningful displays (textual or graphical), but the information relevant to a particular problem is typically well hidden. To make the data useful, one must look at the results of one or more analyses that are run separately or concurrently on the same trace before being combined to provide the appropriate visualization data.

The current analyses we use include profiling information (both single-level, à la UNIX *prof*, and two-level, à la UNIX *gprof*), raw call data, call dags (converting the dynamic call tree into a dag by merging common subtrees), memory-management data, interval analysis where the execution is broken into intervals in which the data is combined, and input/output behavior. While all these are not available with all the current trace packages, we have used each of them with real trace information to address specific behavioral issues.

Our experience here is that more and better analysis methods are required to make better use of the trace data. One approach we are taking is to look at analyses that deal with specific issues such as multiple threads of control (e.g. finding potential deadlocks or race conditions) and message-passing protocols. Another approach is to use techniques developed for data mining to attempt to find patterns, both expected and unexpected, in the trace data. A third approach involves providing a high-level language in which the programmer can define the processing to be done. This last approach is interesting in that such a specification can be used to generate appropriate filters that limit and do some data processing as the trace is generated.

Viewing the Result

Understanding behavior requires not only flexibility in the information to be displayed and the processing to be done on the raw trace data, but also in a variety of different visualization strategies. Different visualization techniques are appropriate to different types of data and emphasize different aspects of the data that they are presenting. Our approach here has been to provide a visualization framework (VALLEY) along with a front end, MIRAGE, to offer the programmer a range of different styles. The current system provides about ten different styles of 3D visualizations including several file-based visualizations similar to SeeSoft, dot plots, compact trees, graphs, time-based mappings, spiral-based linear views, and interval analysis.

The CACTI system interfaces with this framework by letting users select a visualization appropriate to the data they specified. The system analyzes the data and the different visualization strategies to determine which may be appropriate. The user then can select one of the appropriate ones and parameterize it, specifying properties of the visualization and associating data fields with visualization properties.

To be useful for understanding, however, even the most sophisticated visualization needs to be interactive, so that the programmer can browse over the data and then focus on the most relevant aspects. MIRAGE lets the programmer fly over the data, change the various parameters defining the visualization, and, to a limited extent, interact with the visualization.

Much of our ongoing research in this area involves extending these browsing techniques to make our visualizations more useful. Here we are looking into different ways of filtering the data to be visualized before it is displayed, letting the user dynamically control what is displayed or not displayed, and correlating multiple visualizations of the same data in a synergistic manner. Another aspect, work being done jointly with David Laidlaw, involves developing new art-based visualization techniques that can convey more information in the limited screen space available and make patterns in the data more apparent to the viewer.

Conclusion

This research, a continuation of the software visualization work done at Brown over the past seventeen years, has been ongoing for the past six years and we expect it to continue for several more before we have a truly practical system for understanding complex system behavior. The work is funded in part by the National Science Foundation, and we are always looking for additional collaborations. Here we are interested in identifying potential users, establishing research partnerships, getting people's feedback and experiences with software visualization, and identifying specific behavioral problems that visualization might be able to address.

ALUMNI EMAIL

MARY TAFFS '75

I'd like to announce the publication of my first novel, **Martha's Madness**, by Awe-Struck E-Books this September. **Martha's Madness** is a love story at heart, but also the story of a woman learning to value and trust herself. It takes place on the Oregon Coast and in Rhode Island. The hero and heroine met at Brown as undergraduates, and the heroine was a computer science major. I was very pleased with the following recent review which gave the book four stars and classified it a "Must Read." Said reviewer Angie Evans of *Reviews@SimeGen*, "I enjoyed this delightful book. It was well written and the characters were believable. **Martha's Madness** is a good read." A second novel, **Celtic Knot**, featuring a Brown CS grad as heroine will be published in a few months. Both books will be available for sale on diskette as downloads or as Rocket Editions. For more information, visit *http://www.awe-struck.net* or my web site at *http://www.spiritone.com/~mtaffs/index.html*.

I fell in love with the world of books before I could read, and writing is what I've always wanted to do. My computer career started out at the American Mathematical Society in downtown Providence. Later, I worked at SofTech in Newport for several years doing Ada compiler development with my husband Dave ('75). We moved to the Portland, Oregon area in 1986, and I've worked most of the time since then at Mentor Graphics on DOC, technical publishing software which was discontinued as a product in 1991, but is still supported for the sole use of Boeing.

I decided in about 1993 that if I was ever going to write, I'd better get busy NOW. My favorite books at the time were mysteries and thrillers, so I decided to write an amateur sleuth series. I found out, though, that while I love to read them, I'm not good at coming up with all those plot twists and red herrings, and the relationships between the characters were what I most loved writing about. That led me to move more into the women's fiction/ romance area, and that's where I truly belong. *mtaffs@SpiritOne.com*.

MAARTEN VAN DANTZICH '93

Since leaving Brown I've been at Microsoft Research, exploring the use of 3D graphics in user interfaces for desktop applications. I enjoyed visiting Brown in September and presenting an overview of our recent work; it's fun to be the visiting presenter in Lubrano now, and great to see how healthy the department is. In five years many of the people have changed, but it still feels familiar. Not that there weren't any surprises: I found a coffee-cart in the lobby of the CIT, which would have been unimaginable five years ago. Ah, progress!

I enjoy Seattle immensely, and see several former Brown CS folks on a regular basis, including Dan Robbins '91 (who's in the office next to mine and one of my close coworkers), Russell Belfer '91, and Matt Ayers '95/MSc '98. And of course there are lots of Brown alums at Microsoft: enough to fill a whole article with the variety of jobs we have. Maybe in a future ***conduit!***?

JOHN GANNON (BS '70, ScM '72) 1948-1999

John Gannon, one of Andy van Dam's first and favorite students and one of CS's most distinguished graduates, died in his sleep of cardiac arrest at his home in Silver Spring, MD, on June 12. He was 51.

John and his wife Nancy participating in Andyfest last May

John had a congenital heart defect, but surgery as a child had made it possible for him to live an active and productive life. He was Chair of the Department of Computer Science at the University of Maryland, where he earned a reputation as a leading researcher in software engineering and, most importantly to him, as a demanding but caring teacher whose door was always open.

Andy remembers John as exemplifying the best of Brown's "get-involved-up-to-your-eyeballs" undergraduates: he was a UTA, a research assistant, paymaster (he kept the books for the graphics group), helped write proposals and critiqued everyone's work. John had a very idiosyncratic laugh (a wicked cackle) easily evoked; he never took himself or others too seriously, but he had a great seriousness of purpose. He was always reliable and just got stuff done, no excuses—he had a "My dog ate it" stamp made to use on the programs of students reduced to creatively lame excuses for late or incomplete submission.

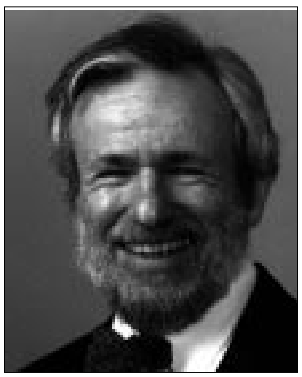
At the time, Andy and Peter Wegner were the two people teaching CS in the Division of Applied Math (John Savage was teaching in the Division of Engineering) and Andy was trying to cover too many aspects at once via AM101 and 102. He included a little bit of parsing and compiler theory, a subject then still in its infancy. One of the things John Gannon and Andy did was to try and make sense of Frank DeRemer's brand-new PhD thesis on parsing to see what of it they could teach to undergrads. John read the thesis and then would try to explain the basics to Andy; Andy would find holes in the explanation, and back John would go to try and work it out. After many such cycles they doped it out and managed to teach hot-off-the-press LR(k) parsing to mere sophomores and juniors.

Doing this digging had a life-long impact on John. He went to the University of Toronto for his PhD, as did many of his generation from Brown, including Ed Lazowska, Frank Tompa, Chris Braun, Larry Weissman, David Elliott, John Guttag, and John Zahorjan—a pipeline from Brown to Toronto that continued for many years. In his professional career, John became a real authority on software engineering and brought rigor to an area sometimes lacking it. He also became a superb teacher and mentor.

The John D. Gannon Scholarship Fund has been established to commemorate John's commitment to students and education at the University of Maryland. Information about this is available via *http://www.cs.umd.edu gannon_memorial.html*. Condolences may be sent to John's family at this email address: *gannon-memorial@cs.umd.edu*. John, who was raised in Rhode Island, is survived by his brother, Rick, of Foster, and by his wife, Nancy Garrison '70, a Yale Law graduate who works at the U.S. Department of Justice. Nancy can be reached at 10108 Day Avenue, Silver Spring, MD 20910.

John's many friends and colleagues at Brown mourn deeply his untimely passing.

THE 23rd IPP SYMPOSIUM



Host Eugene Charniak

The Computer Science Department held its 23rd Industrial Partners Program Symposium (less formally known as IPP day) on April 29, 1999, on "Web-based Natural-Language Technology: Search, Translation and Analysis." As usual, John Savage, who runs the program, warmed up the audience by having everyone introduce themselves and in the process demonstrated that he knew the names of pretty much everyone there (which, from my perspective, is an amazing feat). He then turned things over to me, and I promptly introduced the first speaker (whose name, fortunately, I did know: Bill Woods).

Bill Woods is from Sun Microsystems Labs and has been doing work on using natural-language technology to improve information retrieval. His talk was entitled "Finding What You Really Want: Natural Language Technology in Precision Content Retrieval." The standard technology in this area is to start with the set of words in the user's query and return documents that contain those same words. However, usually users do not want an entire document, but rather just a passage that answers their question. This is the "Precision Content Retrieval" of the title, and it is much harder than document retrieval since the passage now sought contains many fewer words, and thus is less likely to contain exactly the words in the query. Woods looked at how various natural-language technologies can help in this problem. The techniques he considered ranged from the relatively basic to those closer to the research edge. At the basic level I was surprised to learn that "stemming" (reducing a word to its basic form—e.g., the word "reducing" becomes "reduce") has a major positive impact in retrieval rates. Among the more complicated techniques described was "subsumption"—finding concepts that are either more general or more specific than a given concept. The idea here is that if the user asks about, say, "computer" prices, we might also want to give information about "workstation" prices, and vice versa.

The second talk of the day was by Roy Byrd of IBM Watson, on "Text Mining for Knowledge Management." Roy pointed out that one of the most profitable areas of IBM today is its consulting business. This business generates a lot of information about what IBM customers are doing and the problems they are having doing it. This information is both a problem and an opportunity. Essentially Roy and his group at IBM are working on an on-line document creation and analysis package to be used by the IBM consulting division to find documents that relate to their queries, route documents automatically to individuals who could use the information, and use the entire document collection to gather global information about trends in the area. One of the technologies Roy stressed was the use of finite-state automata. Natural language is funny in that few of the tasks involved can be completely solved within the domain of finite-state machines, but a huge number of them can be approximated using these techniques. The advantage of doing so is that finite-state technology is very robust and very efficient. Roy gave some numbers on the speed of processing that were quite amazing. The talk also stressed that the goal here was a system that would be available in the near future but would, at the same time, incorporate a very large number of language tools. The way to meet both of these desiderata is to keep each tool fairly simple (which dovetails with the use of finite-state technology).

Symposium speakers, clockwise from Eugene: Ted Selker, IBM; Steve DeRose, Brown; Bill Woods, Sun; Roy Byrd, IBM

The first talk after lunch was given by Ted Selker, also of IBM, and an IBM Fellow. Ted is known, among other things, for inventing the track-point device on IBM Thinkpads. Ted is also a Brown alumnus and took my AI course as an undergraduate. I invited him because he is well known as an interesting speaker and he has done work that relates to language issues. However, his talk today was not so much on language as on questions about computer design in the macro sense—if one is to walk around with a computer, how big should it be, how should one hold it or otherwise carry it, etc. If the audience ever noticed that the talk did not relate all that closely to the topic of the symposium I am sure they quickly forgot—I know I did. This was the only talk I have ever seen where the speaker came equipped with fifteen to twenty computers and proceeded to pull them out of his backpack and discuss the design issues involved in their creation and why they did or did not work as products. One interesting point Ted made concerned computers used by people gathering interviewing data. It turns out that if the computer is designed so that only the poller can see the screen, people are much less willing to talk than if they can see the screen at least part of the time. More generally, Ted emphasized that one has to think long and hard about how a computer is to be used in the real world before coming out with a product, and that most of the machines he pulled from this backpack never got produced commercially because they failed this test.

The day's penultimate talk was by Steven DeRose, who founded Electronic Books Technology (now Inso eBusiness Technologies), but recently returned to Brown halftime in the Scholarly Technology Group. His talk was entitled "Links, Queries and Language Awareness on the Web." One way to think about this talk is that whereas the other talks were about making computers better consumers of text, this one was about making the computer's job easier by having the human text producer take on the burden of marking up the text to indicate some of its meaning. That is, we want marks that indicate not just how to display some text, but also indicate something about its content. Steve first noted that hypertext markup language (HTML) goes a little way in this direction, but not nearly as far as most people would like. The bulk of the talk then was on XML (eXtensible Markup Language), which has markup symbols for more content types and a standard way to extend the mark set to your domain. One important point here—

I think Steve said it, but if not it was certainly implicit in his talk—is that there is going to be a strong incentive to add these markups to very many web document producers: the web search engines at some point are going to be sensitive to XML markups, and thus if you want them to put your web site at the top of the list for a user's query, you had better put in markups that allow the search engine to "realize" your site contains the information the user has requested (e.g., the price of a new Ferrari).

The last talk of the day was mine, on "The Statistical Revolution in Natural-Language Processing." My goal here was not to illustrate how this technology can be used (I figured that the other speakers would be able to do this far better than I), but simply to show how the use of statistics has revolutionized computational linguistics, and how it has led to remarkably robust and accurate programs that attack a wide variety of natural-language topics. In particular, I talked about four topics: parsing, word-sense disambiguation, pronoun reference, and lexical semantics. I find the work in parsing (assigning a sentence structure to a string of words) particularly exciting. As recently as six or seven years ago, there was no parser that could take, say, the front page of today's *New York Times* and parse every sentence therein, even with numerous mistakes. Today we have several such programs (almost all statistically based), and now the research effort is to drive down the number of errors the programs make. The other topics in the talk were more semantic in nature. Even though statistics are easiest to gather on surface phenomena like words, statistical techniques are beginning to be applied to the less surfacy area of "meaning." So in the work on pronoun reference we took some text, marked it up to indicate the referents of all the pronouns, and then gathered statistics on such things as the probability of the pronoun's antecedent being, say, N sentences back, for $N = 0, 1, 2$, etc. One interesting subproject here relates to the fact that pronouns in English have a gender that must match the gender of the antecedent. Thus a goal was to automatically collect information about the typical gender of objects described by words like "piano" (neuter), "president" (mostly male), etc. This project relates to the lexical-semantics portion of the talk, as the goal of statistical lexical-semantics is to learn semantic information about words by observing how they occur in text.

A recent WiCS (Women in Computer Science) meeting at which Philip Klein was invited to speak. WiCS acts as a resource for women in the department and offers a mentoring program and a 'safe space' for women to discuss related issues

The astaff team is a blur of activity during a stuffing blitz to mail out our graduate recruitment posters. l to r: Fran Palazzo, Genie deGouveia, Lori Agresti and the back of Dawn Nicholaus

NEW CS FACULTY

Vera, Emmanuel and Thomas Hofmann at the CS picnic

The Department is very pleased to be welcoming three new faculty members this year—William Chan, PhD University of Washington, 1999; Amy Greenwald, PhD Courant Institute, New York University, 1999; and Thomas Hofmann, PhD University of Bonn, 1997. Professors Chan and Greenwald won't arrive until the new year and will start teaching in Semester II, so news about them will be forthcoming in the spring issue of ***conduit!*** Professor Hofmann is here and teaching an advanced topics course this semester on information retrieval and data mining.

Thomas and his wife Vera grew up in a town near Cologne. They met in high school and discovered a mutual passion for philosophy. During their time at the University of Bonn, they also spent a year in Paris, one of the great centers for contemporary philosophy, which helped coalesce their future intellectual directions. Despite having taken myriad philosophy courses, Thomas decided upon a career in CS. Vera received her PhD in philosophy. Their respective degrees were awarded within days of each other and they came to Boston, Thomas as a postdoc at MIT and Vera continuing her research at BU. Their son Emmanuel was five at the time and had to learn English fast. He now speaks English very well—he with an American accent, his parents with an Oxford one. The challenge now is to make sure he retains his German.

From Boston, they headed for the west coast and a year at Berkeley—Vera in the Rhetoric department, Thomas in CS. They enjoyed Berkeley's colorful ethnic diversity, culture and climate. From thence they returned to the east coast and Brown. Vera now has a lectureship in philosophy at URI and Emmanuel is now in second grade in Barrington. He especially enjoys the school bus—he's the first student picked up, so he's aboard the longest.

Thomas' research goals can best be summarized by the motto "artificial intelligence by machine learning." He is interested in how computers can take advantage of large amounts of data in order to achieve better performance for a variety of AI-related tasks. His previous work focused on problems in pattern recognition, machine vision, information retrieval, data mining, and natural language processing. Here, the term "data" spans a large spectrum from raw measurements and sensory inputs at one end to discrete and symbolic data at the other extreme. Some of the key questions that fuel Thomas' research are: What are the principles of inductive inference, i.e., how can we infer general laws from a set of examples? What are the mechanisms that would enable machines to understand and interpret images? How can computers process and understand natural language? What methods can be devised to automatically detect structure and regularity in large data sets? How can data analysis and data visualization be combined to make human-computer interaction more efficient? How can computers support humans in solving decision problems in complex and uncertain environments? While he is greatly interested in the mathematics and theoretical foundations of machine learning, he has also a profound interest in modeling and solving specific problems. It is his strong belief that the development of new models and methods can greatly profit from a confrontation with real-world problems, while on the other hand "there is nothing more practical than a good theory" (V. Vapnik). The nature of these problems often requires a cross-disciplinary approach, involving various aspects of computer science from theory to computer graphics to systems design as well as disciplines like statistics, information theory, applied mathematics, statistical physics, electrical engineering, operations research, and computational linguistics. One of his main motivations for coming to Brown is its strong culture of interdisciplinary research and its opportunities for collaboration across departmental boundaries.

Philosophy is still a passion that often finds Thomas reading and arguing into the night. Vera is, of course, a worthy opponent in debate and there's always plenty to discuss. Thomas enjoys writing essays combining philosophy and computer science, analyzing CS's impact on society and the ensuing societal changes. Chess is another major interest. He plays with friends in Germany via the Internet, and thanks to an enrichment program run by the California school system, Emmanuel is also able to play. Vera is an accomplished pianist and church organist. She misses being able to play the organ in her church back home to which she had a key and could practice pretty much at will. Fortunately for all, their piano recently arrived at their new home in one piece.

"One of his main motivations for coming to Brown is its strong culture of interdisciplinary research and its opportunities for collaboration across departmental boundaries"

As a former Green Party activist, Thomas avoids driving his car and opts instead for traveling by bike or public transport whenever possible. He uses the East Bay Bike Path to come to work each morning (as does David Laidlaw). Since he's now in Rhode Island for the long haul, he will likely become involved in politics again.

Tom Doeppner. Tom enjoyed visiting Swami Manohar, PhD '89, and his family last January in Bangalore, India. He was there to teach a course on distributed computing for CS Industrial Partner Compaq.

David Laidlaw. David's new course on interdisciplinary scientific visualization is centered around writing mock grant proposals, reviewing them by emulating the NSF review process and recommending proposals for 'funding.' Practicing what he preaches, David is co-PI on a new KDI grant from NSF, "3D Free-Form Models for Geometric Recovery and Applications to Archaeology," with David Cooper from Engineering as PI. He and Steve Reiss were just awarded an NSF grant for visualizing program structure and execution to help in understanding how large programs work (or don't work...).

Franco Preparata. Franco was recently appointed chair of the Gödel Prize Committee. It is the most prestigious award in theoretical CS for an outstanding paper(s) published in the previous six years. In July he served on an international review committee for the Department of CS of the University of Pisa, Italy. In August Franco presented several lectures at a summer forum on supercomputing organized at El Escorial, Spain, by the Universidad Complutense, in Madrid. Recently he was keynote speaker at a workshop in theoretical CS at the IBM T.J. Watson Research Center.

The Manohar triplets sporting t-shirts from Tom Doeppner

Steven Reiss. Along with David Laidlaw, Steven has been awarded an NSF grant for studying software visualization, as described elsewhere in this issue. He is also teaching CS233, where this semester the class is attempting to build a modern Java programming environment for the Suns. To keep him busy beyond these activities, he continues to work on providing tools for using design patterns throughout the development process, designing user interfaces for mobile computing, developing a front end for searching the Internet with a local startup company, Simpli.com, and harvesting his vegetable garden.

Roberto Tamassia. Roberto gave a keynote lecture at the *VIII Encuentros de Geometría Computacional* in Castellón, Spain. His book, *Data Structures and Algorithms in Java* (coauthored with Michael Goodrich), has reached its 6th printing.

Eli Upfal. In May Eli and PhD candidate Gopal Pandurangan went to the 31st ACM Symposium on Theory of Computing in Atlanta, where they presented a joint paper. In July, Milos Hauskrecht (Eli's postdoc) presented a joint paper with Gopal and Eli at the 16th International Joint Conference on Artificial Intelligence (IJCAI) in Stockholm. He was appointed editor-in-chief of the *Journal of Discrete Algorithms.*

Andy van Dam. Andy co-chaired the first joint European Commission/National Science Foundation (EC/NSF) advanced research workshop, entitled "Human-Centered Computing, Online Communities and Virtual Environments," held in France. As a member of the technical advisory board for Microsoft Research, Andy went to Beijing in June, where Microsoft has recently opened a research lab. Andy was part of a contingent including Raj Reddy (CMU), Ed Lazowska (Chairman of CS at the University of Washington), and others that provided a full day of lectures on the future of computing to some 1500 invited academics and students—the first such gathering in China. Andy's talk was on Post-WIMP 3D User-Computer Interaction.

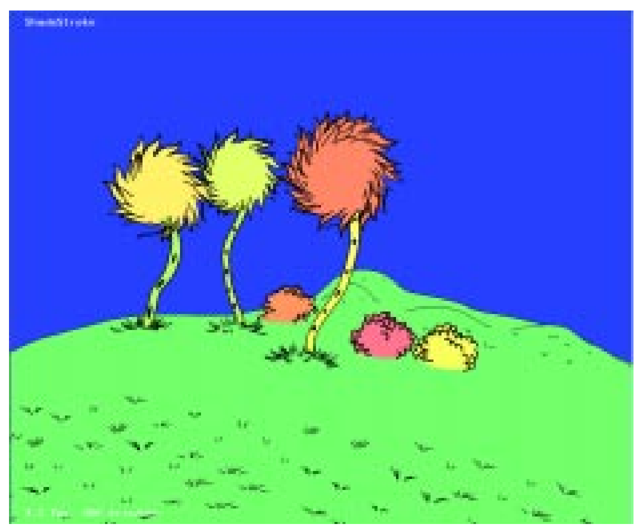
Stan Zdonik. In September Stan completed his duties as North and South American Program Chair for the 25th International Conference on Very Large Databases (VLDB) held in Edinburgh, Scotland.

SIGGRAPH 99 SHOWCASES BROWN WORK

John Hughes

In LA this August about 50,000 people gathered at SIGGRAPH 99 to see what's new in computer graphics. There were tutorials, a spectacular "show floor" with all the latest and greatest new products, panels, a "digital bayou," a film and video show, panels on controversial topics, and technical paper sessions. At least 25 people from Brown attended and participated in several venues. I'm not going to give the details of all of them—but I will describe the technical papers we presented, which is the area I'm most familiar with.

PhD student Lee Markosian was lead author on one paper and coauthor on another, both originating from his work on art-based modeling and rendering, to become his dissertation this spring. The overall goal of this work is to incorporate into computer-generated images the kinds of skills that artists have—representing complexity with just a few strokes of a brush or pen, creating 3D shapes through "constructive drawing," and selectively using detail to direct attention to what's important in an image. Lee's SIGGRAPH papers were "(Gimme some) skin: A constructive approach to free-form modeling" and "Art-based rendering of fur, grass, and trees." The first of these (by Lee, Jon Cohen '00, J.D. Northrup '00, and me) described how to build large and complex free-form shapes (like the torso shown in Figure 2) by a technique based on constructive drawing in which one defines a form by first placing large shapes



This picture, a computer-generated rendering of a Dr. Seuss-like scene, was chosen for the cover of the SIGGRAPH Proceedings. This continues a trend: Brown students, faculty or graduates have been authors of papers featured on SIGGRAPH covers in four of the last five years like cylinders or ellipsoids, cubes, and other "primitive" objects and then constructing a "skin" around them. The skin, in this case, is an adaptively tesselated mesh, positioned to have a user-chosen offset from the underlying primitives in places where a primitive is nearby, and to behave nicely elsewhere. "Nicely" means that it avoids self-intersection and tries to avoid wrinkles or creases, and degenerate triangulations with long, thin, triangles. The speed with which one can create complex free-form models, especially when one has Bob Zeleznik's SKETCH system to help with placing the primitives, is remarkable.

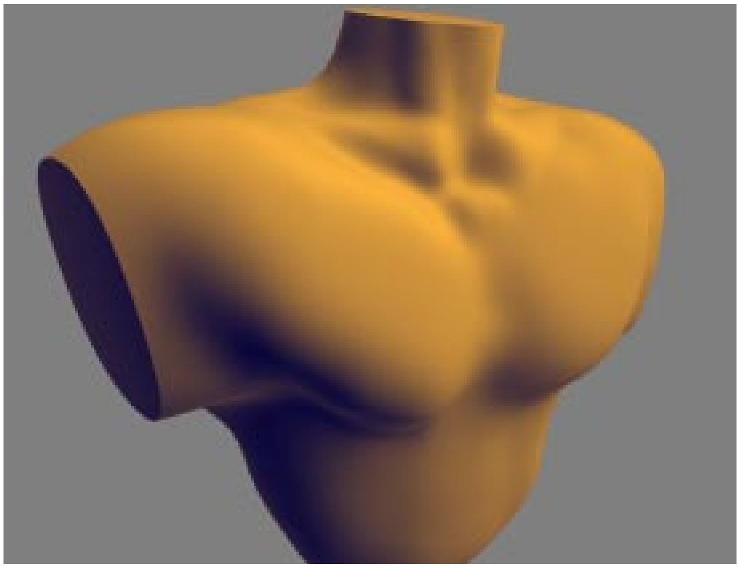


Figure 2: Jon Cohen, who had never used a commercial modeling system and had no background in anatomy, was assigned to produce a torso with our system. His first one looked terrible (it had no collarbones, for example), and I told him to try again. And then again. And then he got an anatomy text from the library, and tried again. And again. Less than 24 hours after he started (with a short sleep break), he produced the model shown here

The other paper got to be informally known in the graphics group as "fuzzy fur feet," because the first successful proof-of-concept picture was based on a drawing from Dr. Seuss's *The Foot Book*. The paper was written by Michael Kowalski (BA '98, Scm '99), Lee Markosian, J.D. Northrup '00, Lubo Bourdev (ScB, Scm '98), Ronen Barzel (ScB '83), Loring Holden, and me. In an early draft of his presentation, he spent the first few minutes explaining why "non-photorealistic rendering" was useful. Fortunately, someone who listened to it said "Michael: if they're there to listen to you talk in the last session of the last day of the conference, I think you can assume that they believe what you did had a purpose!" I'm pleased to say that a lot of people were there—about 1500 of them—and that Michael's presentation was a real success. Our work on this topic isn't done yet: Michael, now working at ATR in Japan, is collaborating with us on extensions to this work and another project on art-based graphics. Look for us at SIGGRAPH next summer!

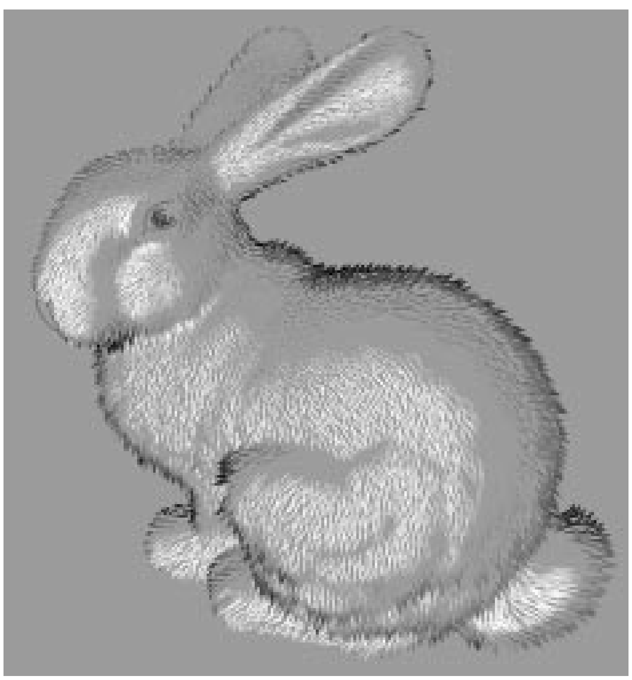


Figure 4: With the same technology used to create the cover image, we also make Durer-like images, with dark and light strokes used to convey highlights and rich texture

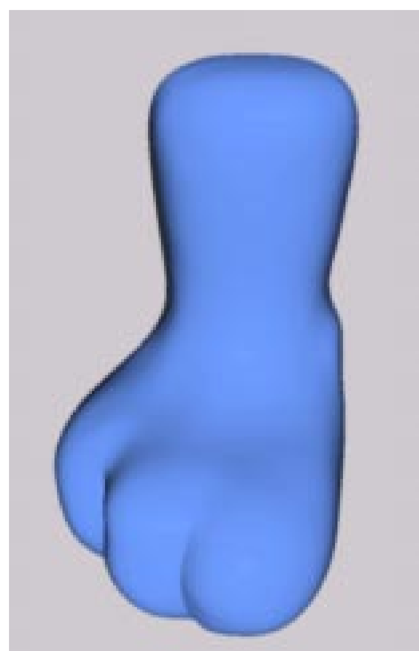
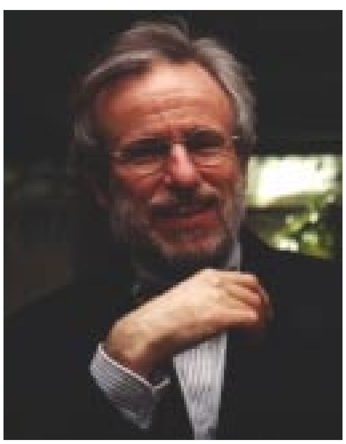


Figure 3: A cartoon-style foot with creases between the toes. This sort of model cannot easily be made with implicit surfaces or with conventional modeling software

Over the summer, President Gordon Gee visited General and Mrs. Kanellakis at their home in Athens to thank them for establishing graduate fellowships in memory of their son Paris and his family and for their support of CS faculty. There was an exchange of gifts—a gloriously illustrated book about Greece from the Kanellakises and for them, a tie and a scarf specially designed as Brown presentation gifts

CHARNIAK UNPLUGGED



Eugene Charniak

Recently a newspaper article appeared in my e-mail on the difficulty universities are having in recruiting and retaining computer science faculty. This got me to thinking about a number of such articles I have seen over the last year.

The first of these was about a year ago, when high-tech companies were lobbying Congress to increase the number of special visas given to foreign workers in areas in which there was a shortage in the US. These companies said that they were not able to hire high-tech workers, particularly in computer science and related areas, and thus it was important for the US to allow more such people to come into the US from other countries. My thoughts on this were somewhat ambivalent. On the one hand, I am proud that the US, as the world's preeminent multiethnic society, can draw the best and brightest from around the world. No doubt this is in part because of the salaries we pay, but I think few would disagree that an equally important reason is the extent to which our culture is open to people of all races and ethnicities. (Which is not to say we are perfect, just much better than most everyone else.) Yet at the same time, I did not like the idea of increasing the visa quotas. As a professor of computer science, I am disappointed that so few American students go into the sciences, and while I do not have any data on this subject, my impression is that despite the rising salaries high-tech workers can command, it is still the case that doctors, lawyers, and business executives outearn computer scientists. If this is true, then the recalcitrant students are making reasonable decisions. Why study something hard, like science, when you can major in something that is much less work and still earn more over your lifetime?[1] Since we live under the sway of the laws of supply and demand, not allowing more foreign workers in should cause salaries to rise even further and, I hope, attract more US students into the sciences.

I was still mulling this over a few months later when I saw a second article, this time about problems the military was having in attracting enough new recruits. As this was a "news" article, the author expressed no opinion on what should be done about the problem, but a rather large number of people were quoted who thought that the best solution would be to bring back the draft. They pointed out several benefits beyond the obvious one, that the military would not have to worry about where their next recruit was coming from. One was that the young would have the experience of helping their country in an important way. Another that I remember particularly was that this would allow the civilian population to better understand military culture, and would at the same time help bring military culture into line with the mores of the average civilian.

1. Our editor, Trina Avery, takes exception to the implication here that science is harder than non-science. Your humble author stands by his prejudices.

I'd like to thank **Shuang Ji** for his very generous donation of $15,000 as an unrestricted gift to the department. Shuang completed his Master's degree working with Steve Reiss in '94. His gift comes at a time when the department is experiencing considerable growing pains as enrollments soar, the number of faculty increases, and the need for equipment and space outstrips the administration's largesse. It's especially gratifying when a former student steps forward to help out the department, and we are particularly grateful to Shuang for his gift and the warm sentiments with which it was given.

Tom Dean, Chairman

...and one last glimpse of Andyfest

With his father, Levie van Dam

Most of the participants pose in front of the Sciences Library

My first reaction was that the author and the quotees were on to something important here, but that they were not thinking big enough. Here, I thought, was the solution to the computer programmer shortage as well. We should draft people into Microsoft, Sun, Compact, SGI, Intel, etc. Not only would this solve the programmer shortage, it would give the young a good feeling for having helped the country surmount a national problem, it would help the civilian world understand the nerd culture, and finally, it might even bring the culture inside of (say) Microsoft or Sun in line with that of the rest of the country.

It was with all this baggage rumbling around in my head (please excuse the mixed metaphor, but it best describes my state of mind) that I read the article on the shortage of computer science professors. As I saw it from my previous reading, there were three ways to attack this problem. Increasing the quotas for foreign professors of computer science would probably not do any good. From my experience here at Brown, there does not seem to be any effective quota here. The typical person we wish to hire has unique credentials, a case can be easily made that there is no person in the US with the same credentials, and, ... well, you get the general drift. This left the other two options: raise the salaries of professors of computer science, or use the draft. On this issue, as you might imagine, I have very firm opinions.

conduit!

A publication of
The Computer Science Department
Brown University

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/

Suzi Howe
Editor-in-Chief

Katrina Avery
Editor

Jeff Coady
Technical Support

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

conduit!

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

Printed on recyled paper

Address changes welcomed