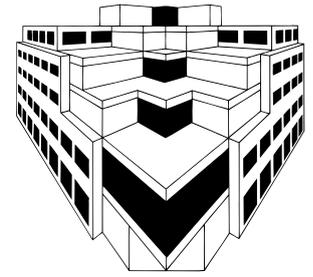




BROWN

conduit!



Department of Computer Science
Brown University

Volume 14, Number 1
Spring, 2005

RESEARCH

Barbara Meier on what
computer animation can learn
from art animation 3

Shriram Krishnamurthi on
security research in access
control 6

FACULTY

Three new faculty join CS 10

Recent books by Tom Dean,
Eli Upfal and Joe LaViola 15

Comments on the fall 2004
Natural Language Processing
Symposium 17

3.4.5.

Tom Doeppner's update on the
CS curriculum 20

Charniak Unplugged 24

ALUMNI/AE

Alums speak out: True tales
from the tech world! 26

Commencement weekend
reunion and alumni discussion
groups 30

Computers and Human Values: A First-Year Seminar in Computer Science

Brown's freshman seminar program initiated three years ago gave faculty in the sciences a welcome opportunity to offer substantive introductory courses that fit the model of a college seminar. Although Brown stipulated only that the new seminars had to be small and populated solely by first-year students, those of us who had benefited from freshman seminars as undergraduates knew that the new courses should focus on important questions as much as recognized bodies of knowledge, and should encourage discussion and open-ended reflection more than memorization of facts or mastery of techniques. For me, the challenge was to create a seminar that helped students formulate their own philosophy of technology through examining recent developments in computer science.

Fortunately, the opportunity to design a seminar came at a time when traditional "computers and society" courses seemed ripe for reconsideration. Taking Norbert Wiener's *Cybernetics* (1947) as the *Ur-text* of this tradition, it's easy to identify texts and courses that have tried to articulate what is at stake, and what should be the subject of debate, in assessing the impact of computing on personal, social, professional, and political life in societies present and future. The rarely acknowledged work of Joseph Weizenbaum and Michael Arbib in the '70s did much to establish this tradition, and since the '80s a significant body of work concerned with computers and social issues has emerged.

It's no surprise that every writer since Wiener has formulated "computers and society" issues differently. What is striking is the degree to which these differences have concerned far more than the nature of available technology. With changes in each generation's understanding of computing's nature and limits have come changing assumptions about the nature and significance of people, cultures, and societies. The result of all these changes has been a series of approaches to computers and society issues, each compelling to the generations that authored them but usually regarded as merely quaint and sometimes incomprehensible by the generations that followed.

The challenge, then, was to design a seminar about computing and human values that today's students would find engaging while motivating the idea that the *questions* inspired by modern computing are not especially new, however different the technological and social contexts. Mindful that 21st-century college students can't be expected to know Wiener from a hot weiner and that when TIME named "The Computer" its "Man of the Year" in 1982, they were not yet born, I wanted to juxtapose writing motivated by recent computer science developments (e.g., robotics, networks, security) with older texts that raise the same questions in different ways. In part, the seminar is an experiment to see whether or not this juxtaposition of new and old sheds new light on contemporary issues.



Roger B. Blumberg

continued on page 9

Notes from the Chair: What's going on at 115 Waterman

Greetings to all CS alums, supporters and friends!

The current academic year has brought major developments to the computer science department – for faculty, students, and the CIT building. As many of you know, ours is not a place of latency or disinterest, and this year clearly illustrates our strong tradition of embracing innovation!

New faculty

This past fall we welcomed three new faculty members: Claire Kenyon, Meinolf Sellmann, and Odest Chadwicke Jenkins. Two of these positions were secured through President Simmons' Target of Opportunity program. This program, which is part of the President's larger plan for academic enrichment throughout the university, provides funding to departments for scholars of unusual depth, originality, and impact whose presence results in diversifying the faculty and ensuring the growth of our educational offerings. We are delighted to have Claire, Meinolf and Chad join the CS family. More on their backgrounds and research appears later in this issue.

New staff

In addition to faculty, the department also hired two new staff members, Lauren Relyea and Laura Zurowski. Lauren, who recently completed an MBA at SUNY Albany and worked as a technical assistant at Hudson Valley Community College, is our new special projects assistant. She will be working closely with graduate recruiting, symposium and event planning, and web communications. Laura Zurowski comes to us from Brown's Office of Institutional Diversity, where she was the special assistant to Associate Provost Brenda Allen. Laura holds a M.Ed. from Harvard and is the department's administrative supervisor and manager of the Industrial Partners Program (IPP).

Undergraduate curriculum

We have also been hard at work reviewing the CS curriculum, especially our introductory courses – see Tom Doeppner's article later in this issue. Our ultimate goal is threefold: to attract and retain the best and brightest students in computer science without sacrificing or diluting educational quality; to increase the involvement of first- and second-year students; and to highlight the interdisciplinary nature of computer science in areas of study such as computational biology, economics and applied math. In this vein, we continue to support student organizations such as WiCS (Women in Computer Science) and external mentoring such as the Artemis Project.

CIT construction

Those of you who have recently visited the CIT building know we have been under construction for most of the year. This large-scale project has been six years in the making and we are very pleased that the university's administration values the improvement of our facili-

ties. Provost Robert Zimmer has been instrumental in supporting our increase in faculty and ensuring that we have ample room. The construction will result in an nearly 50% increase in office, class, lab and study space – including a truly inspirational third-floor open work area complete with whiteboards, good lighting, comfortable seating and tables. This renovation connects the third and fifth floors with a uniform design and, once completed, the center stairwell of glass and steel will not only facilitate movement throughout the department, but also display many pieces from our computer museum collection.

The contractors have promised that all work will be completed by late April, so we hope not to have to distribute hard hats for the CS Reunion and Networking Reception scheduled for Commencement weekend! We look forward to seeing you on May 28th for good music, delicious food and drink, stimulating conversation and a tour of the redesigned atrium.

Eli Upfal is the current CS department chair. A description of his latest book, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, co-authored with Michael Mitzenmacher, appears on page 15.



conduit!

is published twice yearly by the Department of Computer Science and is distributed free to CS alums, faculty, staff, students and industrial partners.

Queries about this publication can be directed to laura_zurowski@brown.edu

Our campus mailing address is:
115 Waterman Street
Providence, RI 02912
401.863.7600

Publisher

Laura Zurowski

Editor

Trina Avery

Faculty Editor

Eugene Charniak

Technical Support & Distribution

John Bazik, Kathy Kirman, Brett Turner & the wonderful folks at Graphic Services!

Computer Animation Can Learn from Art Animation



Barbara Meier

to lose these properties more than those in other media. The possibilities in CG are endless, but many CG animations are more similar to live-action films than to fine art animation, and by this choice the filmmakers are forgoing much of the potential of animation. One of the beauties of animation is the ability to craft individual frames; I find the evidence of this human touch lacking in many CG pieces that are set up by animators and then rendered offline. Of course there are counterexamples in both feature films and shorts including innovative CG, pedestrian art animations, and imaginative commercial work, but I would argue that many CG works could benefit from borrowing ideas from fine art animations in both content and look.

Comparing commercial work to fine art or features to shorts or CG to other media may be unfair given the different goals, target audiences, and processes, but I believe the comparisons are worth making if they can help push the commercial work, features, and CG in new directions. The first CG feature (“Toy Story”) is only ten years old – we are very much in the formative years of this medium. Style and content are certainly matters of individual taste, and I’m not advocating a move toward a specific style. I’m suggesting that CG animation, in general, could experience a wider variety of content and styles, pulling from animation traditions as well as inventing new ones, instead of embarking on a steady slide toward sameness.

Story and storytelling matters

How does commercial work differ from art animation? In creating animations, a filmmaker chooses a concept or story and then translates it into a visual telling of the idea or events. The telling is not the same as the story; the telling is how the filmmaker wants the viewers to experience the story. For commercial work that is marketed to a general audience, both the story and the telling are usually straightforward so that viewers get it, are entertained, and want to

“The Incredibles” was incredible! The story was zany, satirical, and a wild ride. I was utterly entertained by it and by other animated feature films, but I am often more moved by “fine art” animation shorts, typically made by independent animators. They are edgier, deal with deeper subject matter, but, most of all, they express the filmmaker’s vision more purely and directly. These stories and imagery sink into and stay with me for years after I see them. I think 3D computer graphics (CG) animations tend

buy more movie tickets, DVDs, and related toys. But in finding this low common denominator, filmmakers dilute the telling. The sharp highs and lows are rounded off, the imagery is easily readable, and the characters are based on stereotypes. The purity of the concept is lost in the translation.

Fine art animations often do not need commercial success. There is no market for animations as artworks; the “original” film is easily duplicated and therefore isn’t rare and collected like an original painting. Screening fees and profits from tape or DVD sales (if any) rarely cover the cost of production. With no need to please the masses, the artist is free to choose content and the best undiluted “translation.” The story can be edgier and the presentation “messier.” I believe this more direct, sharper vision results in a more heartfelt, personal, and affecting piece.

Clearly financial return is a factor in the ultimate design of commercial work, so making films for narrow segments of society isn’t practical. And I’m not suggesting that CG directors and animators are slackers – it’s clear that most love what they do and care deeply about crafting their work. But I’ve also seen great ideas become either muddled by too many cooks or toned down for wider appeal, less risk of offense, or some other market-related reason. CG shorts like Chris Landreth’s Oscar™-winning “Ryan” show us that CG isn’t limited to “kid stuff.” Could the successful studios take a risk on deeper adult-oriented subject matter? Design characters with complex personalities? Offer a story with subtlety or that isn’t plot-driven?



In “The Dog Who Was a Cat Inside,” director Siri Melchior combined scanned 2D hand-drawn elements and 3D CG models that look like cutouts to create this stylized hybrid look. Image courtesy Passion Pictures, producer and Channel 4 Television.

Understanding differences in process

The animation process in large CG studios differs from that of independent filmmakers. Feature producers and directors oversee hundreds of people who work in an assembly-line fashion. In independent work, the animator is often responsible for everything: the concept, artwork, soundtrack, and editing. This allows him to experiment and even change techniques for different parts of the film. In a small production, many parts of the process are executed painstakingly by hand, sometimes by animating collage elements, paint, or

sand directly under the camera or by using stop-motion techniques with puppets or clay figures. These are risky processes because the animation is “performed” during filming, leaving no record but the film itself. I can’t watch a directly animated film without thinking about the long hours the animator spent getting it just right. I get the same feeling from animations made from hundreds of drawings, knowing that each one has been “touched.” The end result of these processes feels more intimate. As a viewer of CG films, I often feel remote from the process, even though I have done this kind of work myself in a CG studio.

Perhaps it is my very knowledge of the process that makes it feel remote. Just as the story for a film is translated and abstracted by the artist, so is the imagery. The animator chooses what to show: What are the scenes? How are they staged? What is in the background? What is the visual style? In the 3D CG process, the animator starts with dark, empty space that must be filled up. For an interior shot, he builds a 3D room with walls, doors, floor, ceiling, and “stuff” so that when the camera is positioned, the empty black is covered up. Great pains can be taken so that the props and style help tell the story (think of the monster-appropriate accessories in “Monsters, Inc.”). This process is very similar to designing a live-action set.

On the other hand, when a 2D animator makes a drawing, he starts with paper that has some color and texture. If every frame is to be a separate drawing, then the setting is usually very abstracted – perhaps the walls and floor are defined by lines that only indicate parts of edges. Because they are hand-crafted, drawings may appear wobbly when viewed successively at film speed. It would be hard for viewers to read a complex image in which all the parts were moving, so animators may have features come and go as they become important. Other 2D animations are multi-planed: a detailed background remains static while stylistically different animation occurs in front. This kind of simplification and abstraction helps focus the viewer on the important parts of the scene.

Current CG tools, on the other hand, don’t provide a simple way for features to come and go or to change levels of detail in an artistic way. We are stuck with “all the stuff, all the time,” just as in live action. Furthermore, the CG environments invite their creators to create detailed, realistic surfaces and textures. We end up with an environment that is too detailed to be an abstracted version, but not detailed enough in surface and lighting nuances to be cinematographic. Of course, there are exceptions to this. In particular,

the reef scenes in “Finding Nemo” come to mind. The subtle movement of the anemones and plant life made the setting come alive, but the rendering model of the water kept these elements sufficiently in the background that they weren’t a distraction. As CG lighting models become more sophisticated, this kind of subtlety should become more prevalent.

The hand-touched aspect of animation is missing from CG animation in other ways as well. It is easy to create 3D models that are perfect in terms of square corners, smooth curves, and no blemishes. Technicians take great pains to add irregularity and wear and tear to the models and shading, but the results are rarely as convincing as a quick ink and wash drawing on paper can be. Indeed, for CG, the art of translation from idea to visual entity is now a mathematical process of algorithms that simulate movement and light, and the play that happens with the visual image is partially lost. The directness of using a drawing or sculpting

implement has been replaced by manipulating numbers and widgets. Perhaps this is why it feels cold and remote.

I am afraid that choosing the appropriate technique for each story is getting clouded by the wow-factor and the state of the art in CG. Now that fur, cloth, water, and so on can be simulated in CG, it seems they are de rigueur. Does that mean we will never see extraordinary animation effects like the water in Pinocchio’s Monstro the Whale chase scene, or Pocahontas’s flowing hair during the “Colors of the Wind” song? And though I enjoyed the whimsical production design of the movie “Shrek,” I prefer William Steig’s ink-and-wash drawings from the book on which the movie is based. In my opinion, the story doesn’t require flowing capes and swaying grasses, but I’m certain the film’s marketability would have plummeted if created from wiggly drawings.

Drawing from experience – some ideas for the future

Obviously, I have a fondness for some particular styles of animation, but even from a more objective viewpoint, I think some CG animation could be more substantial and inspiring if it incorporated some ideas from artistic animation. What would this look like and what kinds of tools would animators need? That is an open question and I don’t have all the answers, but I’ll propose a few ideas.

When computers were first considered for animation, they were used as “super-photocopiers” to reduce the tedium of redrawing backgrounds, or of applying ink and paint to cels. Eventually, most



In Cassidy Curtis’ animation “New Chair,” 3D CG models are rendered using a simple style (left) that is image-processed to achieve the artistic ink and crumpled paper style (right). Images courtesy Cassidy Curtis ’92.



2D “cel” animation was created by compositing scans of 2D drawings and backgrounds. Even hand-drawn animation is now often created with software that simulates drawing or painting on paper, eliminating the need to scan and photograph, especially in the testing stage. But how might we “touch” frames of 3D animation in a more personal way?

One way I could imagine is to add some details to 3D animations by hand. For example, for a simple character, it might be easier to hand-draw facial expressions than rig the dozens of controls necessary to move the facial geometry of the model and then manipulate them into the correct positions. The hand-drawn part could be direct drawing on the 3D model, or it could be a 2D post-process. This method was used in the CG short “The Dog Who Was a Cat Inside” by Siri Melchior. In addition to a great mix of 2D and 3D models of the cubist-inspired 1950s Parisian set, the animators include 2D animation drawings that appear directly on 3D models.

I think there is a tendency to fill a scene with 3D models because you can, but simplified hand-painted 2D backgrounds can actually create more depth and mood than a complexly lit 3D scene. Hand-animated elements can also add life to otherwise static scenes. In fine art animation, similar drawings are often cycled during static moments to keep the image alive. I know 3D offers versatility in staging and camera movement and much 3D CG animation is intended to look like a fantastic version of live action, but that is just one possible look. When you add 3D and 2D together, you get another whole range of looks.

I’d like to see ways to fill up the empty frame with more abstracted imagery; this can unclutter the image and help viewers focus on the action. A “detail remover” camera could be invented to remove details or objects unimportant to a scene, just as depth of field is used to focus real lenses on specific areas of a live action scene. Another tactic could be to build spare geometry and use it to indicate surface form and texture, just as painters use shorthand strokes to indicate leaves of a tree. And wouldn’t it be great to have an organic drawing-based way to create those simple models? To bring perfect geometry further into the human world, perhaps modelers could have features that automatically “mess up” surfaces with irregularities and wear and tear.

The same idea could be applied to some kinds of motion. Out-of-the-box CG motion is syrupy smooth. It takes a lot of effort to add the noise common in most real-life movements. Changing the rendering style from photorealistic to a drawing-like style could add some of the temporal bumpiness that is naturally present in animated drawings. Some of these ideas have been prototyped, but few have trickled down into production tools. I’d like to see the same rigor that has been applied to making fur look good used to making CG look more hand-crafted.

Chris Hinton, an animator with the National Film Board of Canada, creates his 2D CG working surface by compositing one or more very subtle, nearly transparent textures as his background. Immediately, the expanse of solid color that the software offers has been transformed into a more “physical” surface. This concept could be extended to “unperfecting” many aspects of CG. Going further, we know that computers are used for coloring scanned drawings, but could we do the opposite and have artists color computer-rendered

animated forms? Could computers aid this process without taking it over?

We also need to look for ways to make the CG animation process more direct. By now I should be used to it, but I am still amazed at the insane number of steps animation software requires to do anything, very few of which feel creative. It is important to have access to the controls under the hood for the occasional difficult maneuver, but that is not where I want to spend most of my time. New functionality is important, but I’d also like to see software designers work on intuitive controls that consider the human creative process, not just the underlying algorithms.

The reason I love animation is that anything is possible – any story, any visual style. Animators are not limited by what can be photographed, only by their imaginations. As the commercial animation industry and a portion of fine art animation gravitate toward computer animation, I hope we don’t lose the incredible range of content and visual style possible. As the largest consumers of CG software, commercial studios exert influence on the direction of research and development in new techniques. I hope they expand to new styles, not just refine the existing ones. Similarly, independents should continue to innovate with CG techniques, traditional ones, and the combination of the two as well. Commercial work will never match the extremes of personal expression and experimentation found in the art world, but it can benefit from art’s discoveries and revelations.

Barbara Meier is a Visiting Lecturer currently teaching “Introduction to 3D Computer Animation” CS195-09. She can be reached at bjm@cs.brown.edu.

Salomon Award to Çetintemel and Jannotti

Prof. Uğur Çetintemel and John Jannotti have been selected to receive one of Brown’s highly competitive Salomon Awards. The \$24,000 grant will support work on autonomous sensing and actuation applications. These applications will use sensors to observe the world around them and process the data they gather to reach conclusions and make decisions, and use actuators to affect the world on the basis of those decisions, all potentially without any human involvement.

Autonomous applications significantly extend the scope of today’s sensor networks, which are used primarily to ease data collection for offline analysis by humans. The goal of this project is to develop software abstractions and infrastructures that will simplify the development of efficient and robust autonomous applications.

SECURITY RESEARCH IN ACCESS CONTROL: THE STORY OF A PROGRAM



Shriram Krishnamurthi

I love writing buggy code. Bugs create research opportunities.

In 2002, I was co-chair of a conference called Practical Aspects of Declarative Languages. Like all conference chairs, my co-chair and I agonized about which online conference manager to use. I've never been a fan of CyberChair, which is the default choice for many areas of computer science: not only is it phenomenally ugly, it's also rather buggy and difficult to run.

Given that it's a large and poorly structured Perl application, I felt it would be especially inappropriate given the title of our conference. In a fit of foolishness, I volunteered to write my own. I was initiating a line of research into Web-based application development using Scheme, so what better showcase could there be?

Four months later, it was time for paper submissions to begin. My co-chair wrote me mail asking when the server would be up. Server? Oh dear. But the beauty of it was, thanks in part to our research, the first version of the application took only a weekend to write – and it handled the conference perfectly well. Because my research uses a control structure called the *continuation* and a prior application in this area was called START, and I had only a few minutes to conjure a name before the application went live, I called mine Continue.

Since then, I've been joined by Pete Hopkins (who earned an ScB from Brown and is now completing his ScM here). We've taken Continue from a weekend prototype to a professional product that has been used by several independent conferences. We've even been paid for it! You can find Continue on the Web at:

<http://continue.cs.brown.edu/>

Over these years, Continue has grown to provide several features. One, for instance, is the ability for a reviewer to generate a URL for a subreviewer that gives the subreviewer access to all of the reviewer's privileges – but only for that one paper. Another is the ability for an administrator to change his or her identity to that of a reviewer (akin to the Unix 'su' command). Each of these features is, naturally, quite useful.

Two summers ago, though, Pete found a terrifying bug. While testing the system, he logged in as an administrator, became a reviewer, and generated a subreview URL. When he logged in as the subreviewer, he found he had all the privileges of the administrator! We were fortunate to catch this, but it became clear that we need a more rigorous approach to prevent further such incidents.

Specifically, Pete and I decided to guard the presentation of all data, and of user-interface elements, with *access control* policies.

This is something we perhaps should have done already, because one of the major concerns in a conference manager is the protection of (confidential) reviews from people with conflicts of interest. By applying policies uniformly, we could ensure that we had a comprehensive and consistent approach to avoiding information and capability leakage problems.

In parallel, I had been getting interested in access control from another source. Seth Proctor – another Brown alum, now at Sun Labs – gave a departmental colloquium just before that summer. His talk presented a new access-control markup language called XACML that he and his Sun colleagues were helping develop and popularize. (IBM is implementing a similar language called EPAL.) From his talk and our subsequent discussion, several research questions became apparent. Continue provided the fillip to pursue them.

There is clear value in such research beyond merely one program. The growing use of these access-control policy languages presents manifold conveniences. Administrators can more easily author and examine policies. Developers no longer need to hard-code policies, which are then difficult to trace amidst a program's logic. They can share policies across organizations, potentially creating compound policies out of constituent ones. They can avail themselves of more efficient implementations of policy-decision engines that take advantage of these domain-specific notations. Automated reasoning potentially becomes easier, since it can be conducted over a more specialized and less expressive language.

...Pete found a terrifying bug. While testing the system, he logged in as an administrator, became a reviewer, and generated a subreview URL. When he logged in as the subreviewer, he found he had all the privileges of the administrator!

These benefits are, however, balanced by new challenges. New domain-specific languages invariably burden developers with vastly inferior tool support compared to that available for general-purpose languages. Furthermore, the potential benefits of automated reasoning remain purely hypothetical without actual tools that implement it.

Since this past summer, Kathi Fisler (faculty at WPI), Michael Tschantz (a Brown senior), Leo Meyerovich (a Brown sophomore), and I have been developing a tool suite called Margrave (A *margrave* was a lord or keeper of borders: he was, in effect, a medieval access-control manager) for XACML. The heart of Margrave is a *verification* system. This consumes an XACML policy and a formal property statement and determines whether or not the policy satisfies the property. Indeed, the verifier is structured more generally as a query engine; verification is just a special case of querying. The user can therefore use this component to investigate the behavior of a policy.

While verification is attractive to academics, it's unclear that it is as yet useful for the average user, who is likely to be a system administrator or some other person with little interest in logic and formal statements. Indeed, a more common use-case is that policies get debugged through testing and, when satisfactory, deployed. After some time, either the requirements change or someone discovers an error; either of these leads to a policy modification. The declarative nature of these languages makes it easy to implement the change, and testing will reveal whether the change has the desired effect. However, how will the administrator know what *other* effects it had?

To this end, we have also built a system for *change-impact analysis*. The analysis consumes two policies that span a set of changes and summarizes the *semantic* differences (as opposed to the more easily computed syntactic, or textual, differences) between the two policies. Furthermore, the verification and change-impact analysis components of Margrave are not independent. Margrave has been structured so that the output of change analysis is represented by data structures that can be processed by the query and verification system. As a result, users can not only examine the summary of changes, but also write queries over it and verify properties of it. We believe this will be a particularly useful and powerful mode of user interaction. In particular, properties that are true of a change may not hold of the system in general. This is desirable, since it indicates that we can perform more powerful reasoning about changes than about the policy as a whole.

Margrave currently handles only a subset of XACML. We have, nevertheless, found this fragment useful for writing fragments of policies for real systems. Our working example is, obviously, Continue, for which we have developed a fairly comprehensive set of requirements. We have implemented Margrave using a form of decision diagram as the underlying data structure for representing policies. This leads to a particularly sprightly system; all the policies we have tried to verify discharge in mere milliseconds. We therefore have hope that a tool like Margrave can be used not only to analyze policies after construction (or, worse, post mortem!), but even iteratively and interactively during policy creation.

Users interested in Margrave, or even just in our sample policies, can find these on the Web at:

<http://www.cs.brown.edu/research/plt/software/margrave/>

Access control is a fascinating topic, and we've found several problems in the area that have been resolved poorly or not at all. Studied decades ago, especially in the context of databases and early security research, it has become resurgent in an era of increasing Web-based and other distributed system deployment. To this end, Steve Reiss and I are co-teaching a graduate level course on software security, with an emphasis on access control and information flow. The reading list, available at:

<http://www.cs.brown.edu/courses/cs234/2005/readings.html>
covers both classic and contemporary papers of interest.

Besides its value as a testbed for access control, Continue is interesting in its own right. Academics disseminate ideas, so software that both enables and protects this dissemination is central to our functioning; you might almost consider it "mission-critical". Given that, the current state of conference software is an embarrassment. My goal is to make Continue an example of what a truly dedicated

software engineering effort can achieve in terms of robust and reliable software. After all, if we don't mind our own house, how can we advise others?

Shriram Krishnamurthi is an Assistant Professor specializing in programming languages. He has recently been awarded a Henry Merritt Wriston Fellowship from Brown for excellence in teaching (see below) and a NSF CAREER grant. He can be reached at sk@cs.brown.edu.

Shriram Krishnamurthi Awarded Wriston Fellowship

We're delighted to report that Shriram Krishnamurthi has been awarded a Henry Merritt Wriston Fellowship for the next academic year. Brown University gives this award to recognize (quoting from the award letter) "the distinguished contributions that our faculty make to undergraduate education". Receipt of the award entitles the faculty member to one semester's relief from teaching duties; it is the award committee's hope that "by rewarding your commitment to teaching thus far with a Wriston Fellowship, we will be supporting research that will continue to enrich your teaching in the future".

Shriram's application for this fellowship cited such successes as his last year's CS 190, described in his lively conduit! article (http://www.cs.brown.edu/publications/conduit/conduit_v13n1.pdf). Featuring the design and implementation of a routing system for Brown's SafeRIDE shuttle-bus fleet, the course was designed to stress such real-world skills as dealing with incomplete and ambiguous requirements that change over time, using prototype systems to get a better understanding of the requirements, and dealing with administrative structures whose purpose and thrust are at best orthogonal to the goals of a software project. The application also described his extensive research collaborations with undergraduates, which lead to publication at prestigious research conferences, and his TeachScheme! Outreach program that trains high-school teachers in new ways of thinking about computer science.

Shriram plans to use his Wriston Fellowship to design a new course, provisionally called "Computer Science for Social Scientists", and write his second textbook, *Programming Languages: Application and Interpretation* currently available in draft form from his home page at: <http://www.cs.brown.edu/people/sk/>

DEPTH CUBE



John Hughes

with folks from DepthCube to show off the technology to DARPA later this spring.

What's so exciting about this display? It's a stereo display for which you don't have to wear funny glasses (if you're old enough to remember wearing those 3D glasses in movie theaters in the '60s, don't admit it!), a head-mounted display, or anything else. Multiple viewers can comfortably look at the screen at the same time from a reasonable point of view.

And it's easy to work with – most programs that use conventional displays with Z-buffers need almost no modification to drive the new display. But the really exciting part for us is that a month or two after the talk, Lou gave us one of these displays on long-term loan (see photo).

How does the display work? Inside the box are 20 parallel displays, stacked up one in front of the other, so that the user is looking through a stack of panes. A pixel on any one of these panes can be turned on or off; turning on a deeper pixel makes the object appear to be at greater depth. The planes are multiplexed, so that only one is active at a time; each of them acts as a filter for light that's projected from behind and through the planes.

That sounds as if the depth limit would be just the distance between the planes, but it turns out to be possible to trick the eye into seeing much greater apparent depth. It also sounds as if one could only display 20 depths, but again, there's more to it than that. Just as one can draw a line that appears to pass between two pixels on a regular screen by drawing the two adjacent pixels gray rather than one-black/one-white as in the old days of monochrome displays, one can also make something appear to sit between two planes in the DepthCube by drawing its image on each plane with reduced brightness. This "depth antialiasing" is one of the key ideas in making the technology work.

What's the use of a 3D display? It has applications everywhere, from engineering and scientific visualization (protein structures, for instance, are notoriously hard to grasp from 2D images) to entertainment ("Quake" with true 3D is awesome). The new display

is installed at the Center for Computation and Visualization, and is already being used in volume-visualization projects including Medical Sciences Prof. Kristi Wharton's confocal microscope data and foam-structure datasets from Los Alamos. CS's Jurgen Schulze and Andy Forsberg are also working on how best to display transparent data on the DepthCube.



The LightSpace DepthCube is the world's first solid-state volumetric 3D display. No headgear. No moving parts. Full color. 50 Hz refresh.

John Hughes (a.k.a. "Spike") is an Associate Professor specializing in computer graphics. His recent work includes designing several art-based rendering algorithms, including one for rapid silhouette drawings and another for rendering of fur, grass, and trees. He can be reached at jfh@cs.brown.edu.

Save the Date!
CS Reunion Reception!
Saturday, May 28th, 5-7pm
4th floor atrium, CIT

Join computer science faculty, alums and friends for a reunion and networking reception. Enjoy good music, delicious food and drink, and stimulating conversation while touring the beautifully redesigned CIT atrium.

R.S.V.P. at:
<http://www.cs.brown.edu/events/reunion/>

Human Values ... from page 1

Whether or not they intend to study computer science, current undergraduates regard computing as extremely significant in their personal, social and political lives. This year, when I asked my first-year students what they considered the most important technological development of the 20th century, 80% said “communication networks,” 15% said “biomedical advance,” and none mentioned atomic energy. Leaving aside whether we should wince at this result, it’s clear that college students are ready to think and talk seriously about the personal, social and political significance of computing.

One set of questions that has been with us since Weiner and Weizenbaum, and can indeed be traced back through Marx to Kant, is how to think about the essence, meaning and purpose of human life in an age of intelligent machines. During the Cold War, much writing about human fate in a technological age really concerned the future of humanistic thinking and the future of work; but the old worries about computers and “dehumanization,” not to mention Weiner’s advocacy of “a society based on human values other than buying or selling,” aren’t especially intuitive to many students born in the Reagan era.

So, in my first-year seminar “Computers and Human Values”, we begin with Hans Moravec’s *Robot: From Mere Machine to Transcendent Mind* (1999), a bold, unworried vision of robotic triumph and human obsolescence. Moravec’s book is useful because it both motivates discussions about current robotics research (this year we were fortunate to have Chad Jenkins tell us about his work on autonomous robots) and uses the patterns of biological evolution to argue against human exceptionalism.

Having tried to understand Moravec sympathetically, we then read a book that couldn’t differ more in its perspective or concerns: Hannah Arendt’s classic *The Human Condition* (1958). The students bristle at the difficulty of Arendt’s language, especially after Moravec’s journalistic prose, but Arendt shows that our vision of our place in the world is *historical* and that in the modern age technology always has a good deal to do with that vision. She also gives the students a rich vocabulary for thinking about Moravec’s robotic future without necessarily accepting the personal and social consequences he portrays as inevitable.

We conclude the first unit of the course with a return to the present and another shock in the form of N. Katherine Hayles’ *How We Became Post-Human* (1999). Hayles’ book, inspired in part by Moravec and responding to the work of Weiner, Shannon and Turing, describes the coming “post-human age” in which distinctions between humans and machines, the natural and the artificial, and individuals and collectives have lost their traditional justification and thus appear increasingly arbitrary. For example, in 1960 when the term “cyborg” was first used (by NASA scientists Manfred Clynes and Nathan Kline), there was no question where an astronaut’s “natural” body ended and his “artificial” limb began; but in an age of neuronal prosthetics (and we were lucky enough last year to hear Michael Black talk about such things), it’s less clear how to distinguish the natural and artificial and whether/why one should bother.

The Moravec-Arendt-Hayles unit makes clear that our ideas about who we are and why it matters have a great deal to do with the era and society we live in, so that for us they have a great deal to do

with technology in general and computing in particular. The rest of the seminar uses the same “new-old-new” juxtaposition to examine two other families of questions: the impact of computing on democracies and the impact of computing on the traditions of ethics.

In the second unit, we read Andrei Cherny’s *The Next Deal: The Future of Public Life in an Information Age* (2001), Walter Lippmann’s *Public Opinion* (1922), and Cass Sunstein’s *Republic.com* (2001) to explore what we think valuable and essential in a healthy democracy. This is certainly the least technical of the seminar units, but for many students it shows the impact of technology on society most clearly. Cherny is a contemporary of my students, the youngest White House speechwriter ever when he worked for Vice-President Gore. He argues for a new democracy inspired by the way computer networks have facilitated, and revealed a deep desire for, individual choice in every aspect of American life. His ideas seem to resonate with members of what he calls the “Choice Generation.” However, by the time the students read Sunstein’s description of the “ideological echo-chambers” created by our ability to filter and customize the information and discussions we expose ourselves to online, they recognize the problem as possibly related to the extreme divisiveness of current American politics.

The final unit of the course discusses the developing field of “computer ethics.” We begin with articles from contemporary journals like *Ethics and Information Technology* and *Science, Technology and Human Values* and then read a traditional philosophical work about ethics -- after trying out Kant in the first year and finding not a Kantian in sight, I’ve settled for Smart’s and Williams’ essays in *Utilitarianism: For and Against* (1973). We conclude by looking at current issues in computer ethics, such as privacy in an age of networks and terrorism and the wisdom of ersatz companions like the robotic pets recently come to market.

“Computers and Human Values” satisfies no technical requirements for potential CS concentrators, but it shows the connection between the work we do here and the questions that have defined liberal arts education for hundreds of years. Though I would be happy to have a seminar like this one inspire students to stay in computer science – and each year about a third of my class is already enrolled in either CS15 or CS17 – the importance of these issues, like computing generally, can be observed across the undergraduate curriculum. In that sense, I think of the seminar as important for any first-year student, regardless of the concentration he or she may ultimately choose.

Roger Blumberg is a Visiting Assistant Professor, and is currently finishing a book about computers and education. He also teaches “The Educational Software Seminar” (CS092/ED089), and a brief article about that course can be found on page 19. Roger can be reached at rbb@cs.brown.edu.

Three New Additions to the Computer Science Faculty: their backgrounds, research interests, and why they're so excited to be at Brown

Claire Kenyon *Professor of Computer Science*



Claire Kenyon comes to Brown from the computer science laboratory at Ecole Polytechnique in France, where she has been a professor of computer science since 2002.

Her primary research area is the design and analysis of algorithms, but she has also worked in computational geometry, neural nets, DNA computing and computational statistical mechanics.

Kenyon had a number of offers from other institutions but said yes

to Brown because “she liked the culture we have developed in this department,” including its size and the potential to collaborate closely with faculty and students, according to Eli Upfal, professor of computer science and chair of the department.

Collaboration holds particular appeal for Kenyon, who is the first woman to be named a full professor in Brown’s Department of Computer Science.

“I would rank few pleasures higher than the process of gaining new insights on a research problem, developed from the exchange of ideas during intensive, highly focused work sessions,” she said. “For each of us, research stretches our possibilities to the limit in a joint effort toward the goal of gaining more understanding of the problem under study. In teaching, a similar pleasure comes from seeing a student understand and start to appreciate something new for him or her, particularly when it is some notion which I found exciting myself the first time I learned about it.”

Kenyon’s interest in computer science “was something of a chance event,” she said. A mathematics major as an undergraduate at the Université de Paris, “I had always been particularly interested in discrete mathematics. It so happened that during my senior year, I took a course in differential geometry, which I really disliked. I discovered programming and loved it; I was fascinated by the algorithmic sides of my programming and algorithms course.”

She received the equivalent of a master’s and Ph.D. in computer science from the Université de Paris in 1985 and 1988, respectively. She conducted postdoctoral work at the French National Institute for Research in Computer Science and Control and at the Center for Discrete Mathematics and Theoretical Computer Science at Rutgers, then joined the French National Center for Scien-

tific Research. She also has conducted research at the International Computer Science Institute in Berkeley, Calif. She has taught at the University of California–Berkeley and at Cornell. In 1991, she won a highly respected Prix IBM Jeune Chercheur; in 2002 she was named a junior member of Institut Universitaire de France.

– Tracie Sweeney (reprinted from the *George Street Journal*)

Odest Chadwicke (Chad) Jenkins *Assistant Professor of Computer Science*



If the blockbuster movie *I, Robot* accurately portrays a future where human-like robots serve as collaborators and contribute toward the needs of human society, then Odest Chadwicke (Chad) Jenkins, Assistant Professor of Computer Science, will help make the future happen. Jenkins’ research interests include humanoid robotics, machine learning and computer animation – all areas that could make a cinematic fantasy a possibility.

Jenkins’ work aims to leverage abilities demonstrated by humans in the real world to control robots and virtual characters. His approach involves addressing two major questions: 1) How can human motion be collected in natural situations without instrumentation? and 2) How can mechanisms for robot control be learned from human demonstration and motion?

While doing doctoral research at the University of Southern California, Jenkins realized that existing systems for capturing natural human motion were inadequate. He conceived a new method in computer vision that is capable of extracting both a person’s motion and kinematic structure (i.e., bones and joints) using multiple cameras. His dissertation focused primarily on using machine learning to uncover behaviors underlying kinematic human motion data.

“We’re creating new methods for capturing human behavior and building new robot architectures that will allow robots in the future to autonomously perform higher-level purposeful tasks,” says Jenkins.

The Robonaut, a two-armed, ten-fingered, humanoid robot developed by NASA and DARPA, may be one near-term beneficiary

of Jenkins' research. Since 2001, Jenkins has been among the scientists and researchers from a multi-university collaborative, including USC, the Massachusetts Institute of Technology, the University of Massachusetts and others, that has worked with NASA and DARPA on the Robonaut. Earlier this year, NASA announced it was considering using the Robonaut on a mission to service and repair the Hubble Space Telescope, which would require working outside the spacecraft.

In the fall semester of 2004, Jenkins offered CS148, the Computer Science Department's course on building intelligent robots, which explored the paradigms and problems of robot programming and allowed students to build their own mobile robots.

Jenkins, who comes to Brown after postdoctoral work in USC's Robotics Research Laboratory, notes the collegial atmosphere of Brown and the University's distinctive balance between teaching and research. He hopes to collaborate with computer vision faculty in engineering and computer science as well as with physical and life science researchers working on aspects of brain-machine interfaces.

– Ricardo Howell (reprinted from the *George Street Journal*)

“These problems consist of finding a minimum over a finite set. For computer scientists, these tasks are very challenging due to the large magnitude of the sets under investigation. Frequently, the search spaces contain more elements than atoms in the universe,” says Sellmann.

As a result, sophisticated methods that include linear programming, approximation, efficient data structures, and constraint programming are necessary to reduce the computational effort required to solve the problems these challenges present, he says.

In 2002, Sellmann received his computer science Ph.D. from the Department of Mathematics and Computer Science at the University of Paderborn in his native Germany. He notes that computer science at Brown has a great reputation for bridging theory and practice. He hopes to be among a working group of researchers whose interests include combinatorial optimization, and he looks forward to collaborating with colleagues who have expertise in theory, machine learning and constraint programming.

– Ricardo Howell (reprinted from the *George Street Journal*)

Meinolf Sellmann

Assistant Professor of Computer Science



Meinolf Sellmann says he first approached computer science in hopes of finding an area of study where theory and practice combined in the solution of real-world problems. In fact, Sellmann, who arrives at Brown this fall as an assistant professor of computer science, once planned to become a medical doctor to use theoretical knowledge to diagnose cures.

“When I attended my first course on linear programming, I found my hopes fulfilled in com-

puter science: Real-world problems are modeled mathematically and solved using sophisticated techniques from computer science,” he says. Sellmann conducts research on the borders of operations research, algorithm theory and artificial intelligence.

“I am interested in combinatorial problems as they emerge from and cover a wide range of practical applications,” he says. Combinatorial problems involve allocating limited resources – with a vast set of variables – to achieve desired objectives. Among those that Sellmann has worked on are airline crew scheduling, automatic recording of TV contents, resource management, graph bisection, network design, and the design of scientific experiments.

NSF Career Grant Awarded to Uğur Çetintemel

We're delighted to announce that Uğur Çetintemel has received one of NSF's sought-after CAREER grants. To quote from NSF's web page, “The Faculty Early Career Development (CAREER) Program is a Foundation-wide activity that offers the National Science Foundation's most prestigious awards in support of the early career-development activities of those teacher-scholars who are most likely to become the academic leaders of the 21st century.”

Uğur's CAREER project is called “Datacentric infrastructures for autonomous sensing-actuation applications”; its abstract reads, in part: “Emerging sensing-actuation applications will use sensors to make observations, process the data they gather to make decisions, and use actuators to affect the environment on the basis of those decisions, potentially without any human involvement. These applications significantly extend the scope of today's sensor networks, which are used primarily to gather data for offline analysis by humans.

“The goal of this project is to design and build a datacentric software infrastructure and pertinent abstractions, mechanisms and protocols that simplify the structuring and implementation of robust autonomous applications operating on sensor-actuator networks. The system will enable developers to specify application-specific data acquisition, processing, routing, and actuation logic using a unified data-centric framework. The system will employ both generic and application-specific optimizations to improve its operation, as these specifications are logical and do not precisely indicate where, when, and how execution should take place.”

Michael Black

Michael Black had the usual flurry of conference and workshop travel in the last few months with the most interesting destination being Montevideo, Uruguay, where he gave two invited talks at the International Symposium on Representation of Reality by Brain and Machines. Michael found Montevideo a vibrant, and very European, city and his hosts provided the best of Uruguayan beef, wine, and music.



In the fall he also traveled to Oxford where he gave a talk to the Robotics Research Group and stayed in New College, where his hosts had access to the very fine wine cellar. His most recent trip was to England's Lake District to give an invited talk at a Rank Prize workshop on Machine Understanding of People and Their Responses. The workshop was located in the town of Grasmere, where Wordsworth is buried, and which is an ideal spot for walking "the pastoral steeps / that shine inverted in the deep / of Grasmere's quiet vale."



Michael's group received a gift from Intel Corporation to support research on human motion tracking. In addition, he and his neurotechnology colleagues received funding from the Veterans Administration for "Rebuilding, Regenerating and Restoring Function after Traumatic Limb Loss" and received an additional \$1M of support for a continuing Office of Naval Research grant studying human motor responses.

Last spring Michael taught a graduate class on "How to build a 3D person tracker" in which students each built a computer

program to track human motion in video sequences. This sort of human tracking was a basic research problem just 3-4 years ago and the students' programs were close to the state of the art.

Eugene Charniak

Eugene Charniak is, even as he types this, in Prague. He was invited to give three talks in this year's Mathesius Lecture Series at Charles University in Prague. His first lecture was rough going because he completely misgauged the audience. He was expecting advanced computational linguistics students, but in fact, the lectures attract a very wide group of students, most of whom have had no statistical background at all. Thus all the really nice equations he had typed into PowerPoint got tossed out, to be replaced by an impromptu lecture on conditional probabilities, Bayes' Law, and finally getting the audience to the point at which they could understand the first slide of the talk.

John Hughes

Spike, having returned from his sabbatical in France, has been trying to settle back into work; he's on the SIG-GRAPH papers committee and papers advisory board again this year but, as you see from the photo, he's also been out playing this winter: that's him on his iceboat with his son, Jack – barely visible except for his hand on the tiller – sailing on South Watuppa Pond in Fall River in February.



Chad Jenkins

In spring 2006, Chad Jenkins will be teaching CS196-2, "Innovating Game Development", a course emphasizing gaming innovation through application of advanced topics in computer science. In preparation for this course, Chad attended the Game Developer's Conference this March in San Francisco with ScM student Jason Mallios. Despite the game

developers' interesting definition of "embodied autonomous agents", Chad is totally stoked about focusing his course on experimental gameplay and applying CS research to video-games.



Shriram Krishnamurthi

Shriram spent most of December in Europe. He and Kathi attended a workshop at Dagstuhl, made a mini-tour of Weihnachtsmarkts both tiny and huge, and went on to nearly two weeks in Switzerland. The highlight of the trip was three days in dazzling conditions in Mürren, in the heart of the Swiss Alps. This was Shriram's first excursion with his (first!) digital camera. He hasn't lugged his SLR in years, but it took him little time to realize how much he missed photography. This semester Shriram is vagabonding in an undergraduate course on the modern research university and is trying to learn new (human) languages.



Barbara Meier

Barb has been hosting informal animation screenings on Fridays at lunchtime. Recent screenings have explored the (often failed) quest for love, antics of cats and dogs, strong women characters, and works animated directly under the camera with

sand, paint, and clay. Few of these animations are available to general audiences in theatres or via video rental; they are typically viewed only at animation festivals. She's hoping to expose her students and colleagues to works they wouldn't otherwise see and to share her love for this medium.

Barbara wishes to thank Jeff and Max for installing a fabulous surround sound system for use during the screenings; everyone who attends enjoys the full theatre experience! All are welcome to drop by on Fridays.

Meinolf Sellmann

Meinolf joined the faculty last September and started teaching this spring. He developed a new course, "Introduction to Combinatorial Optimization" that he is currently teaching. Recent conference travel brought Meinolf to Toronto to the Constraint Programming Conference. In research he is currently working on shorter-path constraints, symmetry breaking in constraint programming, approximated consistency for automated recording, and optimization under uncertainty. You can also find Meinolf practicing and performing with the Brown University Chorus in his spare time.

Meinolf is also preparing the spring IPP symposium, "The Future of Combinatorial Optimization", to be held on May 4, 2005. He is "very glad that we could get a whole group of very distinguished speakers from industry and academia together in the same building on the same day!"

Eli Upfal

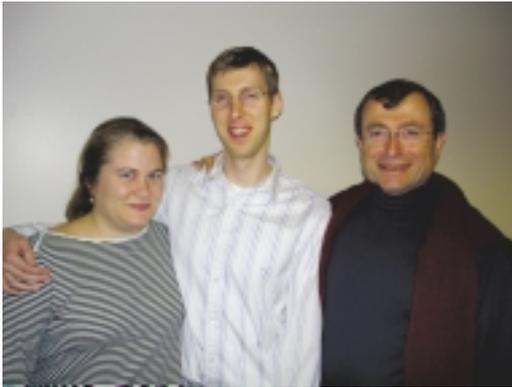
Eli Upfal was the program committee chair of the 45th Annual Symposium on Foundations of Computer Science (FOCS 2004). This is one of the two top annual conferences in theoretical computer science, with publication impact far beyond any journal in the field. The committee had to select no more than 63 papers out of around 300 submissions – a nontrivial task encompassing two full days of meetings and a long evening session. The big reward was attending the conference in Rome in November. Late fall is a perfect time to visit Italy; the hot summer and the tourists are gone, and one can enjoy a long walk in the city with a good espresso on Via Veneto.

After the Rome conference, Eli stayed in Italy for a small workshop on models and algorithms for information networks. This meeting took place in a thirteenth-century castle in Bertinoro – a small, peaceful town on the hills not far from Bologna now turned into a nice conference center. While the technical talks were very good, the evening grappa was even better.

F
A
C
U
L
T
Y
N
O
T
E
S

Andy van Dam

Andy has been collaborating with Anne Spalter on a new course, CS0024: Visual Thinking/Visual Computing. This spring course was funded by the Atlantic Philanthropies grant overseen by President Simmons that targets new and innovative undergraduate courses. The course is innovative in its intensively multidisciplinary approach, its use of technology in teaching, and its focus on an often marginalized area in the academy, namely visual thinking and communication. The subject matter lies at an intersection between art and science and explores the newfound technological emphasis on visual thinking in modern cultures. The course's goal is to lead students to an understanding of how computer-based images can be created and interpreted so they can incorporate this knowledge in their chosen disciplines.



Anne Spalter, Jeff Beall, BA '96, ScM '98, of PDI Dreamworks, and Andy after Jeff's presentation on "Shrek 2: The Art, Business, and Technology of Manufacturing CG Features" for CS0024.

Additionally, Andy was elected Fellow of the American Association for the Advancement of Science (AAAS), lectured at the 2004 Aspen Symposium held by the Forum for the Future of Higher Education, and published "Next-Generation Educational Software: Why We Need It and a Research Agenda for Getting It" in the March/April 2005 issue of EDUCAUSE Review (Rachel Becker and Rosemary Simpson, co-authors).

Stan Zdonik

Stan was elected to the Board of Trustees of the VLDB (Very Large Databases) Endowment, a non-profit organization "promoting and exchanging scholarly work in databases and related fields throughout the world."



Tom Doepfner surveys the construction on the third floor. When completed, the third through fifth floors will be united with a redesigned stairway. The third floor will also have faculty offices, conference space and a well-appointed kitchen.

Recent Research Awards

Black, "CRCNS: Learning the neural code for prosthetic control", DHHS, \$335,594, 8/04-5/05

Cetintemel, "CAREER: Datacentric infrastructures for autonomous sensing-actuation applications", NSF, 6/05-5/10, \$491,000.

Herlihy, "Transactional memory", NSF, \$319,927, 8/04-7/07

Krishnamurthi, "Lightweight analysis of program evolution using feature signatures", NSF, \$146,656, 9/04-8/06

Krishnamurthi, "CAREER: Formal verification of aspect-oriented software", NSF, \$400,000, 9/05-8/10

Laidlaw, "ITR: (ASE)-(sim+dmc+int): Computational simulation, modeling, and visualization for understanding unsteady bioflows", NSF, \$450,000, 10/04-9/05

Savage, "NIRT: Technologies, architectures and performance analysis for nanoelectronics", NSF, \$1,300,000, 8/04-7/08

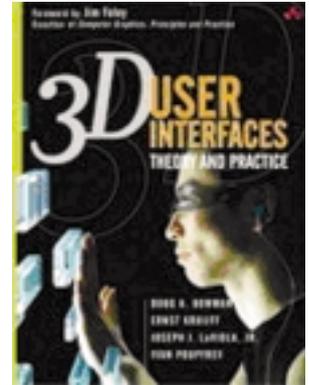
van Dam, "ITR: Collaborative research-(ASE)-(SIM): building biologically based immune system simulations for education, etc.", NSF, \$160,000, 10/04-9/06

3D User Interfaces: Theory and Practice

By Doug Bowman, Ernst Kruijff, Joseph LaViola, and Ivan Poupyrev

Addison Wesley Professional, 2005

3D User Interfaces: Theory and Practice addresses the critical area of 3D user interface design – a field that seeks to answer detailed questions that make the difference between a 3D system that is usable and efficient and one that causes user frustration, errors, and even physical discomfort. The authors present practical information for developers, the latest research results, easy-to-follow guidelines for the UI designer, and relevant application examples. While there are quite a few books devoted to user interfaces in general and to 2D user interface design in particular, 3D user interfaces have received significantly less attention. The results of work in the field are scattered throughout numerous conference proceedings, journal articles, single book chapters, and websites. This field deserves a reference and educational text that integrates the best practices and state-of-the-art research, and that's why this book was created.



Joseph J. LaViola, Jr. is currently a Ph.D candidate in the Computer Science Department working under the direction of Andy van Dam.



Steve Reiss stands next to his newly granted IMB P655, which will reside in the Internet Lab. Luckily, Steve didn't have to move it from the loading dock to the Lab: "it's as big and heavy as an old VAX".

NSF Career Grant Awarded to Shriram Krishnamurthi

We're delighted to announce that Shriram Krishnamurthi has received one of NSF's sought-after CAREER grants. To quote from the NSF web page, "The Faculty Early Career Development (CAREER) Program is a Foundation-wide activity that offers the National Science Foundation's most prestigious awards in support of the early career-development activities of those teacher-scholars who are most likely to become the academic leaders of the 21st century."

Shriram's CAREER project is called "Formal verification of aspect-oriented software"; its abstract reads: "Module systems have grown significantly in scope and sophistication. The most recent innovations have been in the space of so-called aspects, which provide modularity mechanisms that blur the line between static and dynamic composition. The creation of innovative module mechanisms gives programmers new powers, but in turn also makes it possible for them to introduce ever more subtle errors into software. This potential for new kinds of errors places a greater burden on verification techniques. These techniques have, however, failed to keep pace with advances in software modularity.

"This proposal will advance the state of research in computer-aided verification for the forms of modularity introduced by aspects. It will generate new theories of modular verification that address the different styles of aspect modularities. This work is, therefore, an instance of a larger research program that seeks synergies in the confluence of software engineering, programming languages and computer-aided verification."

Natural-Language Processing Fall 2004 Industrial Partners Program Symposium



Eugene Charniak

The 33rd IPP day, on last October 29, dealt with natural-language processing (NLP). We in this field are in the delightful position of finding ourselves more and more relevant (thanks to the Web), and better and better able to live up to others' expectations of what we can do for them (thanks to the last 15 years of progress from the statistical techniques that are ever more prevalent in the field).

The first part of this last assertion was well illustrated by the second talk of the day (I will return later to the first) by Peter Norvig of Google, also a Brown graduate, class of '78. Peter is director of search quality for Google, and his talk was an overview of research on NLP at Google. Of the ten or so projects he described, the one I remember most clearly had to do with word clustering.

One way to find words that have similar meaning is to cluster them by the words they appear close to. For example, "car" and "truck" are, as words go, quite similar in their meaning and appear next to lots of words in common, like "get into" or "the red". Words like "lettuce" and "concert" do not share these similarities. It has been known for some time that if one clusters words in this fashion one usually gets somewhat reasonable groups, like "car, truck, vehicle, motorbike, cab, tank," etc.

What one does not get, however, are the names of these groups, i.e., "vehicle" for the above cluster. Many years ago a researcher at Berkeley suggested looking for phrases like "cars, trucks, and other vehicles" and five years ago a student and I applied her idea to this problem. Peter described a project at Google that started from the same observation but applied it to Google-size corpora and got really good results. In general, the talk brought home the idea that, when applying statistical techniques, having truly massive amounts of data really pays off.

To get back to the first talk of the day, I led off the day with an unscripted (I was too lazy to make slides) talk on the basic techniques that would underlie most of the rest of the day's talks. Probably the highlight of my talk was a very loud (but friendly) disagreement with Salim Roukos, head of IBM's NLP group and another of the day's speakers.

I was talking about machine translation, and in particular the noisy-channel model. This model comes from the early days of information theory, when it was used to understand how to recover messages that have been corrupted by a noisy communication channel. In that guise the noisy-channel equation looks like (where m is the message and c the corrupted version)

$$p(m|c) - p(m)p(c|m)$$

The import of this equation is that if one knows the corrupted message, one way to find the most likely original message is to find the message that (a) is a likely message (makes $p(m)$ high) and (b) could have been corrupted easily to produce the observed message (makes $p(c|m)$ high).

From the viewpoint of machine translation (MT), one imagines that, say, the speaker intended to speak English, but the message got corrupted and came out in, say, French. The problem is to recover the "uncorrupted" English. This sounds pretty weird, but it works extremely well. Salim and I were arguing about WHY it works so well. I still don't know if the audience was intrigued by two experts shouting at one another or bored by what they saw as an abstruse disagreement.

The third talk of the day really illustrated how well the noisy-channel model works. This talk was by Daniel Marcu, a researcher at ISI and also a senior person at Language Weaver, a three-year-old company selling MT software. Their software is generally considered among the best in the business, and is all based upon this model. Daniel showed a video of their work in translating an Al-Jazeera newscast. Unfortunately I had to miss it because I had to



Symposium presenters included: (top row) Daniel Marcu, Mark Johnson, Peter Norvig, (bottom row) Salim Roukos, Bob Moore, and Eugene Charniak.

teach then; however, my colleague Mark Johnson (the last speaker of the day) was there and has told me how impressed he was. Mark is a pretty hard-headed guy, and like all of us old-time professionals in this field can detect film-flam from miles away, so his endorsement is a great compliment.

After lunch Bob Moore of Microsoft gave a talk on an interesting combination of MT and named-entity recognition. The problem in named-entity recognition is to identify a particular string of words as referring to, say, an organization (e.g., Brown University) or a chemical (DNA) or, in the case of the documents Bob was

concerned with, the name of some part of the Microsoft operating system or user interface (e.g., “the View Source Table button”).

There are many reasons for wanting such entities explicitly noted in the text and, interestingly, machine translation can help in this task when the entities are explicitly noted in one language but not the other. So, it is typically easier for automatic methods to find them in English because most of the words will be capitalized; in French it is harder because typically only the first word will be capitalized. Bob showed that if one happens to have the named entities indicated in one language, this can be used (a) to find the corresponding named entities in the other, and (b) to improve the overall translation. This is one of those projects that I like because once one proposes it it seems an obvious thing to do. And furthermore, it worked, which many “obvious” projects do not.

The penultimate talk of the day was by Salim Roukos. Salim talked about so much that it is hard to summarize – the basics of statistical MT, IBM’s Arabic MT system, results on the use of parsing in MT (a topic I am currently looking into), information retrieval when the request and document are not in the same language, etc. I am going to use Salim’s discussion of the basics of statistical MT to include a little more basic education into this article.

“From the viewpoint of machine translation (MT), one imagines that, say, the speaker intended to speak English, but the message got corrupted and came out in, say, French. The problem is to recover the ‘uncorrupted’ English.”

As Salim pointed out, the basic building block of statistical MT is the “parallel corpus”, a body of text for which one has both the original and its translation into a second language. Typically most sentences in the original will be translated into exactly one sentence in the second. One can automatically detect most exceptions by look for things like paragraph boundaries (which typically occur at the same points in both texts) and cases where the sentence in one language is, say, twice the length of the possible corresponding sentence (which may mean that it really is two sentences in the translation). This process is called sentence alignment.

Then one starts on word alignment. This is much harder because unlike sentences, words are often moved around during translation. This is where the noisy-channel model mentioned earlier comes in.

The last talk of the day was by my colleague Mark Johnson in which he described some joint work he and I have been doing on the problem of speech repairs. Frequently in speech people hesitate and then rephrase something they started to say. (“I need a uh want a ticket to Boston.”) Transcriptions would be greatly improved if such mistakes can be removed.

Interestingly, this too can be modeled using the noisy-channel model mentioned above, where now the “noise” is the speech error. In our most recent work we have found that a language model (the $p(e)$ term in the above equation) based upon a grammar of English works quite well here.

At first this might seem odd, since the material overall is

quite ungrammatical. On the other hand, its ungrammaticality should cause a syntactic model to assign such sequences very low probability compared to the same sentence without the repair. This in turn might aid in correcting for them. In his talk Mark outlined our experiments that show this to be the case.

The last event of the day was a group discussion with all of the speakers except Peter Norvig, who had had to catch an early plane. Like many such discussions, most of the content was enjoyable but forgettable and, by me at least, forgotten. However, Bob Moore made one comment that I thought at the time was quite interesting and has proved quite prescient: that automatic paraphrase was going to be important in the future. In particular, he guessed that Google’s news digest was still in beta version because Google did not know yet how to commercialize it without getting into copyright trouble with the newspapers it quotes. Just a week ago Agence France Presse filed suit against Google, and Google has dropped them from their page. So the day ended with a much better than usual discussion.

Industrial Partners Program: Advantages of Membership

Opportunities for close collaboration with faculty and students on research projects.

A guaranteed place at Brown’s spring and fall Career Fairs; fees are waived for the spring event. Partners are invited to join CS faculty at the Networking Brunch.

Invitations to present talks in our IPP Seminar Series.

Invitations to fall and spring IPP Symposia on topics of interest to Partners.

Invitations to all Computer Science seminars and colloquia. Faculty and students are also available to give talks on their research interests at Partner sites.

Employment and internship opportunities posted year-round at prime locations throughout the Computer Science building. Employment and internship opportunities are also coordinated with Brown’s Career Development Center.

Subscription to our departmental newsletter *conduit!*, distributed to all Brown CS alumni and students, chairs of the leading CS and CE departments in North America, and many others, some 2,500 people in all.

Visit: <http://www.cs.brown.edu/industry/ipp/> or contact Laura Zurowski (lpz@cs.brown.edu), Michael Black (black@cs.brown.edu), or John Hughes (jfh@cs.brown.edu) for more information

CS STUDENTS AND FACULTY: PUTTING EXPERTISE TO USE IN THE COMMUNITY

FROM THE CLINK OF A COIN
TO THE CLICK OF A MOUSE

A Rhode Island couple will realize their vision of changing people's lives - one mouse click at a time - when a new online philanthropy site created by a handful of Brown students goes live this spring.

The concept for the site, *Enrichanother.com*, comes from Pat Mastors, a journalist for WPRI Channel 12, and her husband, businessman Jim Mastors. The couple envisioned a robust, one-stop Web site for charitable action, whether that be a business looking to fund a worthy community project; a nonprofit seeking a direct way to appeal for funds; or a youngster wishing to donate an outgrown bike to a child in need.

This past summer, the Mastorses approached CS faculty in hopes of securing a sponsor for the public service project. Steven Reiss was teaching a new freshman seminar, "Building a Web Application," in the fall. Working with the Mastorses, he realized, would give the students the experience of building a commercial-scale Web application for a real-world client while learning the basics of software engineering and the technologies necessary to create sophisticated Web sites - such as HTML, Perl, PHP, and databases.

Reiss brought his class on board. What followed was a semester full of readings, lectures, discussions, and demonstrations, including a November 24 presentation to Governor Donald Carcieri and his staff.

The course "was nothing at all what I expected," said Andrew Chin, who began the class anticipating such assignments as building a Web site calculator or writing a file upload page - good exercises, but "not really practical," he said.

On December 9 the students presented a working version to the Mastorses, who were thrilled not only by what they saw - a complex system of integrated information presented in an easy-to-use format - but also by the dedication and talent demonstrated by Reiss and his students.

Though the class itself has ended, several students continue to work with the Mastorses to refine the site and develop it further. "To just leave the project now, after all this work, would be rather dissatisfying," said Chin.

- Tracie Sweeney (reprinted in part from *Inside Brown*)

SOFTWARE SEMINAR
HELPS AREA EDUCATORS

"Building a Web Application" isn't the only computer science course in which students work for the public good. Since 1990, students in CS92/ED89, "Educational Software Seminar" (created more than a decade ago by Andy van Dam and now taught by Roger Blumberg) have collaborated with teachers in Rhode Island public and private schools to create custom software for use in their classrooms.

Although the course is offered during the spring semester, preparations begin in the fall when Blumberg solicits requests for proposals from teachers. Students in the Seminar review the ideas, select the ones of greatest interest, and divide themselves into project teams. Throughout the semester, the teams work closely with the teachers, their pupils, and others to design and develop software that entertains as well as educates.

Last May, CS92/ED89 students demonstrated three interactive software programs developed during the spring semester. One is a vocabulary game called *Word Expander*, created for kindergarten teacher Ellen Lynch at the Vartan Gregorian Elementary School in Fox Point. This game uses Lynch's idea of "word families" to teach and reinforce spelling and word recognition. *Bridges Through Time* was created for Betsy Hunt, a third-grade teacher at the Lincoln School in Providence. She sought an interactive program she could use to enhance the unit she teaches on

the history of bridge design and construction. *Real Reading for Real Readers* is a comprehension program developed for Martin Carruso, who uses it to prepare his eighth-graders at the Nathan Bishop Middle School for Rhode Island English Standards tests.

There are more than two-dozen programs created since 1998 that are available for downloading through the course website - you can even find some of the Hypercard programs from 1992-1997 there! Included are programs for Macintosh and Windows computers, as well as several Web-based applications. All of the programs are documented on their respective project pages. Visit <http://www.cs.brown.edu/courses/cs092/cs92.download.html> for more details.

- Tracie Sweeney (reprinted in part from *Inside Brown*)



Students in Steve Reiss' class work with Pat and Jim Mastors to develop Enrichanother.com.

3.4.5.**News and Views from the Computer Science Department****Java, C++, and the CS Curriculum***Tom Doeppner*

We in Brown CS have long prided ourselves on producing students who are not merely well educated in computer science, but are also excellent practitioners who snap up top jobs in industry and fit immediately into the most demanding research and development groups. From our beginning programming courses to advanced courses in systems, we've always had an intense curriculum with a heavy emphasis on project work. We've done our best to let students work with state-of-the-art tools and environments and have them use the most appropriate programming languages available.

Doing all of this entails frequent curriculum changes. A year and a half ago we began discussing such a change, a seemingly minor one: changing the programming language used in one course. All hell broke loose. This article attempts to explain what was proposed, what the reaction was, and what we're doing about it.

The course in question was CS32, "Introduction to Software Engineering," taught by Prof. Steven Reiss. Steve's intent was to bring the course into the 21st century by using a language that's in keeping with where modern software engineering is heading. He chose Java, a managed language, to replace C++, an unmanaged language, which had been used in the course for a decade or so. The change in programming language engendered much discussion among the faculty. Among our concerns was that some of us teach advanced courses for which students do projects in C and C++. Where would they learn these languages? It was not at all clear how to resolve these issues.

Somehow this discussion leaked out to students and alumni/ae. Another topic the faculty was discussing was how to involve alums in department affairs. Suddenly a fair number of alums were involving themselves. Those who saw the need for this change probably didn't trouble to speak out. Not everyone saw the need, though, and many of these non-seers definitely spoke out. Here's a sampling of some of the mail we received.

I was told that there was a debate about whether to use Java or C++ for CS32. If you are at all interested in hearing my opinion then please read on ... [we were very much interested!] I think it is debatable, but quite possible that more entry-level software development is done with Java than with C++. However, teaching CS32 in Java would, in my opinion, be a huge mistake for the

following reasons: ... [A number of reasons were given, including the use of C and C++ in advanced courses, the need for understanding programming at a lower level so as to understand the workings of processors, and the belief that there are more jobs available for C++ programmers than for Java programmers.]

I heard on the grapevine that the CS department has decided to stop teaching C++/C in CS32 and instead teach Java. Is this true? As someone who has recruited heavily from Brown, and watched schools get dropped from our recruiting schedule because Java replaced C/C++ as the basic language of instruction, I am deeply concerned. [This alum went on to say that in his company his group preferred Brown grads over all others because they:]

- are smart;
- have a solid understanding of computer science;
- have practical experience writing large amounts of software in C/C++.

I also want to express concern about the department's trend away from C and C++, and the decision to teach primarily managed languages [such as Java]. Unmanaged coding is not dead! The project I am currently working on is being developed in entirely unmanaged code, as are many projects across [my company]. There are days it seems like managed code is dead, but not unmanaged. Furthermore, when I was interviewing for a job a little over a year ago, basic proficiency in one or the other of these languages [C and C++] was something every company I talked to was looking for.

It looked at first as if we might have a real dilemma on our hands – one camp insisting that Java is essential and the other camp being equally insistent that C and C++ are essential. Before we see how all this was resolved, let's take a quick look at the history of programming-language use in Brown CS. (This discussion relies on the possibly faulty memories of some long-term faculty – please send me any corrections!)

Back at the dawn of time (1965 or thereabouts), there was Andy van Dam, who taught assembly-language programming, first on the IBM 7070, then on an IBM 360. The first (and only) CS course sequence available to undergrads at the time was AM101/102. Later there was AM100 and then AM51, which was the intro programming course and the ancestor of the current CS15. By the time I arrived in 1976, Andy was teaching PL/1 (my favorite programming language at the time) in AM51. PL/1 was beginning to fall out of favor (there never was a PL/2!). Within the academic community, Pascal (the language, not the professor, who arrived later) was becoming popular. Though "state-of-the-

continued on page 23

WiCS: Celebrating Women in Computer Science

In October 2004, four members of the Department of Computer Science's Women in Computer Science (WiCS) group — Sarah Bell, Sara Hillenmeyer, Danielle Karr, and Stacy Wong — traveled to Chicago to attend the 2004 Grace Hopper Celebration of Women in Computing entitled “Making History.”

From the press kit, the Grace Hopper Celebration of Women in Computing is “a world class technical conference for women in the field of Computer Science providing a forum to inspire, educate, encourage and create awareness of opportunities for women in the field of computing and to celebrate the considerable achievement of women in the field.” The celebration is named for Admiral Grace



photo by d.karr

Danielle Karr, Sara Hillenmeyer, Stacy Wong, and Sarah Bell celebrate in Chicago.

Murray Hopper, one of the pioneers of computer science. She joined the Navy WAVES (Women Accepted for Voluntary Emergency Service) in 1943 during World War II and was charged with the programming of the Mark I electromechanical computing machine. She went on to invent the compiler in 1953. The most well-known image of Hopper is her warning her students to “remember your nanoseconds” while brandishing twelve-inch lengths of wire, indicating how far an electron can travel along the wire in a nanosecond. Her vivid push for coding efficiency – and general excellence – embodies the positive spirit associated with the Grace Hopper Celebration.

In addition to touring the lovely city of Chicago, Sarah, Danielle, Stacy, and Sara attended a variety of panels, lectures, and presentations organized by the conference. Notably, three events featured Brown alumnae. Sarah Allen ('90.5) of Laszlo Systems gave a talk on “Designing the Next-Generation Web UI in a Declarative XML Framework.” Seema Ramchandani '02, Sc.M. '03 moderated the panel “Choosing Industry and Still Studying: Recent Graduates Share Insight into the Reality of the First Years of a Corporate Career,” and Katrina Ligett '04 and Rachel Weinstein '02 participated in the panel “The Role of Mentoring in Recruiting and Retaining Female Undergraduate Students in Computer Science.”

The Grace Hopper Celebration also included opportunities to meet representatives from Microsoft, IBM, Google, and other top

companies, something welcomed by the three seniors in the group, Sarah, Danielle, and Stacy. All attendees made sure to participate in the social events, dances, and party that made the conference a true “celebration.”

The Grace Hopper Celebration is held every two years, the next one to be held in October 2006 in San Diego, CA. More information can be found at www.gracehopper.org.

Other WiCS events over the past year have included lunches with Google engineers, our very own Jen Rosenbaum '04 from Teach for America, and new department member Chad Jenkins. Most recently, WiCS sponsored “Kabob and Jobs,” a dinner during which current seniors (and alums Kit Colbert '03 and Miriam Goldberg '04) shared insights about the job search and interviewing process. Future events include a bagel lunch with department newcomer Meinolf Sellmann, as well as a meeting to discuss course registration. The semester's events will culminate with the annual senior brunch.

If you have questions, ideas, or want to get involved with WiCS, please e-mail wics_coord@cs.brown.edu.

– Danielle Karr '05

Robert Redford is (Like) a Robot

In “Sneakers”, Martin Bishop (Robert Redford) gets thrown in the trunk of a car and taken to meet his best-friend-turned-nemesis Cosmo. Later, he tries to figure out where the meeting took place. He doesn't know where it was because he was in the trunk of a car and didn't see where they went. He did, however, hear what was going on during the ride. With the help of his friend Whistler (a blind man), he successfully recreates his journey based on what he heard, and goes on to save the day and win the girl.

Put another way, Bishop uses his sense of hearing to recreate a path through physical space. He is able to separate out the important sensory input from all the background and assign them to real-world places. This skill turned out to be quite useful for him, as it would for a robot as well.

“But Dan!” you cry, “Robots don't get kidnapped and driven around in the trunk of a car!” Quite right, but they do have to make sense of a lot of sensor input to maneuver in the world.

One of the areas of robotics research that deals with this issue is simultaneous localization and mapping (SLAM). In SLAM, a robot tries to make a map and locate itself in it at the same time. Maps usually take one of two different forms. Metric maps record the locations of everything in a space but take up a lot of memory. Topological maps simply record connections between different subsets of the space, taking up less memory but still allowing useful things like path-planning. Think of the maps near the elevators of the CIT: these are topological maps since they show the different regions of the floor and how they are connected.

To create a topological map, the world needs to be divided into regions. Every space in the world thus needs to be assigned to a region class. This is equivalent to answering the question “What kind of space am I in?” For humans, answers may include such classes as “Room”, “Hallway”, “Closet”, and “Atrium.”

Robots, however, can have different answers because robots have different sensors. Humans have five senses that they use to figure out the space class they are in. Robots, instead, must use such things as sonar, lasers, radiation and chemical detectors. These sensors react to different signals in the environment, so that robots perceive the world differently. Robot space classes, then, must be distinguishable to these sensors.

Robots are often taught to classify space in the same way as humans. That is, humans try to describe how a specific class, such as “Room”, would look to the robot. I think it is better to let the robot develop its own space classes, based on its own sensor capabilities.

To do this, we view the robot’s sensor input as a high-dimensional space, with each set of readings a point in that space. So, for a robot with 24 sonar sensors, each set of readings is a point in 24D space. Given a set of readings, we use nonlinear dimension reduction to find a lower (5-10) dimensional manifold that fits this data. This embedding is a transform of the data that captures almost all the pertinent information. We then use Bayesian clustering to discover sensor data classes in this space. That is, we take physical space classes to correspond to a Gaussian distribution of sensor readings in the embedded space.

Once we have these classes, it is a simple matter to decide the class of space from which new readings come. New sensor data is taken from the high-dimensional input space into the embedded space and then classified according to the Gaussian model we learned.

These classes need not correspond to the human classes for the same space. As an example, we ran our system on sonar and infrared readings from a section of the CIT’s fourth floor. While we break the area up into two basic classes, our method shows that the robot can actually distinguish six distinct types of space. As a test of utility, we took new readings from the same places and reclassified them using the classes we learned from the first data set. We’re pleased to say that readings from the same physical space were classified the same, showing that the robot can answer the question “What kind of space am I in?”

– Dan Grollman is a current Ph.D. candidate

An Internship at Pixar is Truly “Incredible”

This winter break, I received word from Pixar Animation Studios that I had been accepted for their technical director internship program. I began the internship on January 24th and so far have been at Pixar for about two months.

As a technical director intern, I’ve been placed to work on Pixar’s film due in 2007 (after “Cars”) in the Global Technology department. Global Technology works on developing technology specific to a film (whereas the Tools department might work on technology used on several or all projects). My work has primarily consisted of developing and optimizing the character models the animators will

be working with. Pixar is doing lot of interesting things with the character models in this film that will make the characters appear to react more with each other and their environment. It’s been very interesting to see the system of defining deformations in a character and building the models for animation.

Everyone I’ve met at Pixar has been very friendly and very smart. It’s sometimes intimidating to walk down the halls and realize that a lot of the names by the doors are names I’ve seen on many papers in computer graphics. Even so, there’s little pretentiousness and everyone is very open if I come to them for advice or help.



photo by e.chang

Jack-Jack joins in the celebration for the two Oscars awarded to ‘The Incredibles’.

I usually work with other technical directors but have also had the chance to work with some animators as well. The technology my department develops is usually to the needs and requests of the animators. The animators provide regular feedback on the tools they use and as a result the tools become tailored specifically for what they need. Collaboration between both the technical and artistic sides of computer animation is key, and both sides meet frequently.

Aside from work, I’ve also had the opportunity to take a class in figure sculpture in Pixar’s “Pixar University” program. Classes are offered in art and film as well as technical training, allowing employees to expand beyond their own fields. After sitting in front of a computer screen all week, it’s nice to be able to exercise yourself artistically.

Part of the fun of being at Pixar has also been being here for the Oscar celebrations. After “The Incredibles” won two Oscars, Pixar celebrated by having champagne for all of its employees in the atrium while Brad Bird and John Lasseter recounted their stories from Hollywood. It was great to be part of the celebrations and be there when all 800 employees lifted their glasses to congratulate one another on their success.

Pixar has been a great place to work at; being here makes clear how its unique environment has made of each of its films such a success.

– Edwin Chang ’05.5

Curriculum ... from page 20

art” Pascal became available on the VAX/Unix system we acquired in 1979, we couldn’t handle all the students in CS11 (the number given to what had been AM51 when CS became a department in 1979). So CS11 continued to use PL/1 on the university’s IBM mainframe. In September 1983, however, we leapfrogged all other CS departments in the world and began teaching CS11 on Apollos, the first commercially available computer workstation (Three Rivers Corporation had announced their Perq workstation earlier, but Apollo was the first to ship a product). No one else had enough such computers, nor the naïve guts, to do such a thing. We were at (or beyond) the state of the art.

Pascal evolved over the next two decades and, by the time Pascal the professor (Van Hentenryck) arrived, Andy was teaching object-oriented Pascal and CS11 had been renumbered CS15. We’d switched from Apollos to Suns. Eight years ago Andy switched again, this time to Java, first on our Suns and now on our Linux PCs.

So, everyone now learns Java as their first language. But no, that’s not quite right. In the mid-’90s we realized that an intensive programming course might not be the best first course for everyone, not even CS people. So Leslie Kaelbling and Philip Klein got together and produced a new introductory sequence of two courses, CS 17 and 18. Students learned Scheme as their first language, and then ML and a bit of Java. Now students have a choice: they can start with CS 15, then take CS 16 on algorithms and data structures, or they can take CS 17/18.

But where did students learn C and C++? While we all agreed that neither language was a good first programming language, it had, by the mid-’80s, become essential in some of our advanced courses. Students either learned C on their own or took mini-courses offered by the department at the beginning of the semester. Steve Reiss began teaching C in CS191, “Software System Design,” in 1983. Though 191 was aimed primarily at seniors, CS132, “Introduction to Software Engineering,” first taught by Stan Zdonik and then by Steve Reiss, was introduced in 1985 and was required of all concentrators. Finally, everyone was learning C. Seeing that software engineering was moving not just to object-oriented programming but to C++, Steve switched 132 to C++ in 1989, several years before the language became really popular. Few decent tools were commercially available for it at the time, so the class used a debugger written by Steve himself. In 1993, CS132 became CS32. More recently, C++ has also been taught in a “mini-course” at the beginning of the semester for those CS123 students who haven’t yet taken 32.

Fourteen years after pioneering the use of C++ in the curriculum, in fall 2003 Steve again anticipated the trend in software engineering and moved CS32 to Java. But computer science has become a larger, more complex discipline. While in the 1990s people doing operating systems, networking, data bases, graphics, etc. used the same language as software engineers, this is no longer the case. We’ve thus really had to come to grips with the facts that students doing serious software engineering should be working with managed languages such as Java and C#, but that it’s still important for many of our students to learn and use C and C++.

How did we resolve the C++/Java dilemma? As a temporary fix, we introduced a new half-credit course: CS34, “Introduction to Systems Programming” teaching C and C++, ably handled in spring

2004 by Ph.D. student Manos Renieris. We couldn’t find one faculty member who could carry the course on this year, so this spring it’s being co-taught by Pascal (the professor) and me.

But, as those who have taken CS169 since 1996 can attest, a half-credit course doesn’t really do much towards satisfying graduation requirements and must be taken in addition to all the other courses students need for graduating. So, starting in spring 2006, we’ll replace CS34 with a new, full-credit course: CS36, also called “Introduction to Systems Programming,” which will introduce a number of systems-programming topics as well as teaching C and C++. I’ll be teaching the course and am still putting it together; a tentative course description may be found at <http://www.cs.brown.edu/courses/cs036/>. We’ve modified our concentration requirements to accommodate this new course: wherever CS32 has been required, students may now take either CS32 or CS36. We feel that students who are really serious about programming should take both courses, so we’re allowing both courses to be taken for concentration credit for the Sc.B.

We’ll undoubtedly be making further changes to our concentration requirements in the next few years. Though it might be nice if such changes wouldn’t bring about as much controversy as the Java/C++ debate, it’s been great getting feedback from our alums. Keep that email coming!

Tom Doepfner is an Associate Professor (Research) and Vice Chair. He is the current Director of Undergraduate Studies for the Computer Science concentration, and as chair of the Space Committee has been overseeing the recent renovation. Tom can be reached at twd@cs.brown.edu.

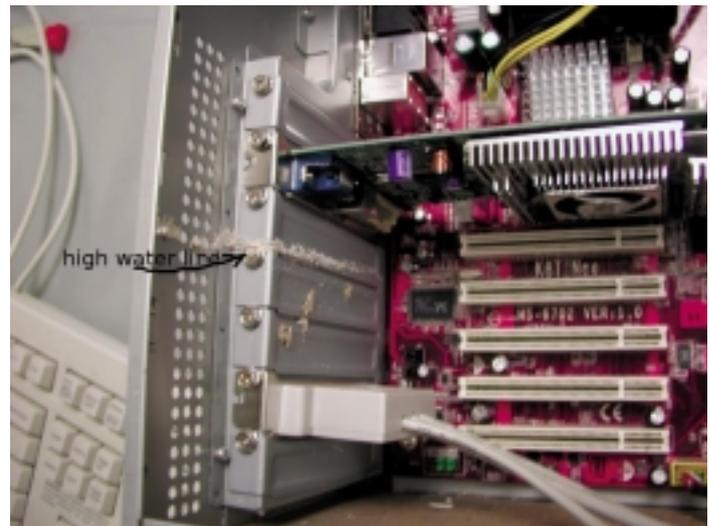


Photo taken after a water pipe burst in January, leaving several offices with substantial flood damage. Despite being partially submerged, this PC was still operational. That’s “Max-Built” quality!

CHARNIAK**UNPLUGGED**

About the only “Unplugged”-type story to hit the CS department over the last six months is the big flood. It started out as the blizzard of '05 – Providence got 25 inches of snow, 50-mph winds, and a day off from school. I got a flooded office. It seems the grad students next door to me left their window closed but not fastened, the 50-mph winds opened it up, a water pipe froze and burst, and the next thing you know several offices, mine included, were under several inches of water. About the only thing faintly amusing about the story is how I learned of it. I was sitting at home (no way was I going to work that day) and was logged into my machine at work. One of my emails stated that several machines, including the one I was logged into, were under water and were going to be shut down! It is a testament to the quality of Max-built machines that they can stand up to punishment like that.

Other than this, there has been a dearth of amusing and/or outrageous goings on lately, at least here in CS. Elsewhere there has been a bonanza. For example, a tenured professor in the ethnic studies department at University of Colorado has made news lately by calling 9/11 victims “little Eichmanns”. However, the Colorado ethnic studies department has not called me up to write a column for them, so I will forgo heaping my scorn on the guy. (Some unsolicited advice for U. Colorado – don't try to fire him. Instead, consider disbanding his department to prevent still more tenured academic problem children.)

Then the mid-eastern studies department at Columbia has been accused of antisemitism by several Jewish students. This one is harder to judge from a distance, by which I mean, of course, on the basis of what I read in the *New York Times*. The *Times* article makes it pretty clear that the department faculty are all arabists, and from my knowledge of Ivy-league professors I would be very surprised if any of the faculty are right of center or even very close to the center. But from this to antisemitism is a long way. Certainly all the faculty quoted in the article strongly deny antisemitic feelings, and I am inclined to take them at their word. At the same time, I believe that the students came by their feelings in good faith. Very early in my faculty career I learned that students take a professor's comments more seriously than the professor does. I remember the first or second time I taught my artificial intelligence course I said that I was about to lecture on a topic X and was going to assume that the students already knew Y, which was required to teach X. I asked the students how many knew Y. Then for the heck of it I asked how many knew X. When one or two raised their hands, I jokingly said they shouldn't be taking a course that they already knew. Of course, the students took it seriously and I had to reassure several of them that I was only kidding and they were really welcome in the course. I now try to use only real jokes in my course. (Do you know the one about the rabbit

typing outside his burrow when a coyote comes along?) At any rate, my guess is that at Columbia several of the professors said things that the students took more seriously than intended. These things happen.

The last and biggest academic dustup is at Harvard, where much of the faculty is in revolt against the president (of Harvard, not the U.S.; if it were the latter it would not be news). The mess was precipitated when President Summers was asked to make some provocative comments at an off-the-record conference and had the stupidity to talk about the status of women in the sciences. The

question was, and is, why are women represented in the physical sciences at rates much lower than in the general population? Summers threw out several mostly non-controversial possibilities – women may be overtly discriminated against, the culture is biased against women in the sciences so they themselves decide against it, etc. Unfortunately for him, he also considered the possibility that women are, on average, less talented at mathematics than men. This last offended many

women at the conference, and one stormed out of the talk (I think it was just one, but as usual I refuse to do real research when writing this column, so you will just have to trust my memory).

This controversy spilled into Harvard Yard and from there into the news media. Unfortunate, almost all the debate seems to take the question as a political one, whereas, at its heart, it is an empirical one. That is, when asked, most academic men and women respond with political comments to the effect of “this is just the usual thing of putting women in their place” or “if Summers really thinks this, Harvard should just stop admitting women.” I am hardly an expert on the topic, but in my guise as an adjunct professor of cognitive science I once read an article about it so I will tell you that my impression is that there a 50-50 chance that women really are, ON AVERAGE, slightly less strong than men in math. On the one hand, the results supporting the contention strike me as reasonably solid. The differences between men's and women's spatial abilities are well documented, as is the relation between spatial and mathematical abilities. Furthermore, the arguments I have heard for the contrary position strike me as quite weak (e.g., there is a low correlation between mathematical ability and interest in the physical sciences. Probably true, but it is not clear how this even relates to the issue). So why do I say only 50-50? Because there is one political response to which I do resonate, and that is the one pointing out that we have been down this road of assuming innate difference to explain societal differences many times before and have been wrong, to a first approximation, 100% of the time. Jews are less intelligent, Asian are less intelligent, Italians are less intelligent, etc. With that sort of track record, one has to be cautious.



One last point before putting this topic, and my column, to rest. Steven Pinker, the cognitive scientist, has written a very good article on this controversy (see his web site). He too thinks the issue of mathematical inequality is a valid one and may well be true. But he goes on to ask why, when confronted with this question, so many otherwise intelligent and rational people give sputteringly political responses. His answer is quite interesting. In a psychological experiment students were asked to give arguments for or against a controversial idea such as awarding adoption permits to the highest bidder. None could, or at least would, do so. Instead they simply said something like "That's obscene" and repeated it some number of times for emphasis. Pinker points out that this behavior is not irrational. All societies have taboos, and if one wants to be a member of the society, it is not wise seriously to consider breaking them. To use Pinker's example, the correct response to "Would you consider selling you spouse and children?" is not "Well, how much?"

The moral of the story is simple. If you are the president of a university you no longer have free speech. In particular, do not even consider stepping on taboos. Of course, from the uproar you would have thought Summers was proposing sleeping with his sister, but the principle is the same.

Eugene Charniak is a Professor of Computer Science and a founding faculty member of the department. He is never at a loss for words. Eugene can be found at ec@cs.brown.edu.

ASTAFF & TSTAFF NEWS



Suzi Howe

With this issue conduit! says goodbye to its Editor-in-Chief of 14 years, Suzi Howe, and welcomes Laura Zurowski in her place. Suzi retired from Brown last January to enjoy, among other things, her beautiful grandson Gio, her almost as beautiful garden, and her interest in photography. Her going-away presents to astaffs were some of her handsomely matted and framed flower and butterfly photographs.

Suzi's retirement caused many sad faces in the department, and we were extremely lucky to find Laura to fill the large hole Suzi left behind. Laura has been at Brown since 2000 and comes to us from the office of Brenda Allen, Associate Provost and Director of Institutional Diversity. She is now astaff's supervisor and conduit!'s new Editor-in-Chief. Goodbye, Suzi and welcome, Laura!



ASTAFF: (left to right) Dawn Reed, Genie deGouveia, and Jennet Kirschenbaum.

IT FEELS LIKE A LONG TIME...
This January, several astaff and tstaff members received longevity awards at the University's B.E.A.R. celebration. As they are all so youthful in appearance and spirit, we won't disclose their years of service!



TSTAFF: (top) John Bazik, Jeff Coady, Max Salvas, and (bottom) Dorinda Moulton.

TRUE TALES FROM THE TECH WORLD — CS ALUMS TELL ALL!

The Challenges of Making Better, Cheaper Computer Animated Films

Jeff Beall, '96, ScM '98
DreamWorks Animation

Successful filmmaking is about collaboration, effective decision making, and a shared vision for what the end result will be. Making a computer animated feature film is especially difficult since everything about the look of the film has to be designed from scratch. And of course, the big question is whether or not the current technology will even allow for the film's characters and the world they live in to be realized as pixels on the screen without costing a fortune to produce.



Anne Spalter and Jeff Beall chat with students after Jeff's guest lecture, "Shrek 2: The Art, Business, and Technology of Manufacturing CG Features".

This last issue is of particular concern to me. I am the Studio Pipeline Architect for DreamWorks Animation, which means it is my job to figure out what about our production process can be streamlined and abstracted away so that our artists can focus on the creative business of filmmaking instead of on the technology itself. Where are the artists fighting the computer or their understanding of the process instead of implementing the creative feedback they received from their supervisors? How much of their time is spent on legitimately difficult creative tasks instead of working with difficult software? The problem is that the tasks involved in feature animation are so complicated that the answers to these questions are not obvious. Because of the demanding creative goals of our films, many new techniques need to be invented to achieve a certain visual effect or animation performance style, and each new technique raises the overall level of complexity of the production process.

We want to be able to continue to push what is possible with computer animation so we must solve the problem of managing the increasing complexity of the production process. One solution we are pursuing at DreamWorks Animation is to leverage our technology base across multiple films so that a technique invented for one

film can be easily reused on several others. We are also working to improve how our "assets" behave in our pipeline. An asset can be a character, a piece of set dressing, or a visual effect like fog or fire, for example. We'd like for our assets to be smarter, such that if you add an asset to a shot, it will automatically know how to reconfigure the shot's data so that the asset will end up in the final rendered frames. Again, the goal is to make our software do all the tedious stuff so that the artists can focus on making the best images possible.

It's a great time to be in the computer animation industry. Even though it has been ten years since the first feature-length computer animated film was released, there are still plenty of challenges that the industry as a whole must work through. Since Moore's Law alone will not allow us to keep up with the kind of imagery that our creative visionaries want to make, we need to rethink our filmmaking process so that we can do more with what we have today.

Proto

Byron Binkley '02 & Jeb Boniakowski '02
Proto

Attendees at the 2002 Computer Science Department graduation ceremony may remember that it was announced that I planned to move to the South China Sea to become a pirate. Attendees may also recall that Byron Binkley '02 planned to continue at Brown and become a Master of Computer Science, and those who were fortunate enough to attend the '01 ceremony as well may recall that Finnegan Smith planned to become a hobo and to that end had relocated to Wyoming. As frequently happens to people in their early 20s, our plans all changed. Byron enjoyed a successful, though brief, career on Wall Street, opting for retirement in 2003. Finn and I were also unable to resist the siren song of New York City, and through a series of twisty passages of fate, all alike, the three of us and a recent Stanford CS graduate ended up living together in a former leather-goods factory with no rooms or kitchen.

Having come into a supply of techy-looking office furniture at the irresistible price of free, the four of us decided the next logical thing to do was to start a technology company. Fortunately, Byron had been working toward that end since leaving Wall Street and had the vision mapped out: we would build a system to enable regular people, you know, people who don't get references to vintage text adventure games, to build their own software.

Of course, end-users write programs every day, primarily using spreadsheets like Microsoft Excel. Byron had extensive first-hand knowledge of the state of end-user software development from his time on Wall Street, and his interest in the problem is one of the reasons he left. Wall Street financial analysts often have strong math and science backgrounds but not necessarily strong programming skills. Many of them manage to build very complex programs using only spreadsheet primitives. However, the spreadsheet paradigm, lacking

support for even the most basic complexity-management techniques like user-defined functions, is ultimately very limiting. To better serve their needs, we've designed a dataflow visual programming language optimized for their problem domain.

Byron was able to secure seed funding from a former Wall Street colleague on the strength of an early technology demo, and since then we've been working intensely on building the product and the business. The eleven of us who are now Proto Software, Inc. released an alpha version of our product in February to very positive feedback. The strength of the alpha helped us find interested testers at a few major investment banks, and we are on track to release a beta in June. Version 1.0 is scheduled to ship in August. Designing a dataflow visual programming language for end-user programmers in the financial industry touches on a lot of different areas of computer science: programming system design, human-computer interaction, interactive visualization, distributed computing, and computing theory. If anyone is interested in talking more about these topics, Proto Software, or if you're interested in abusing our beta, please send an email to jeb@protosw.com.

Industrial Espionage

Ben Sigelman, ScB '03,
Google

What I needed was a *spy*. Exactly two years ago (to the week) of this writing, I was a senior at Brown, and I got an offer from Google. I remember being excited, of course, and also nervous about the decision that lay before me: not so much about why I would go to Google instead of another tech company, but why I would work at a tech company in the first place. After the requisite parental consultations and some hard thinking in the shower, I decided that I would give the for-profit tech world a shot for a few years and then make a principled decision about my career path based on what I'd learned. So, in the spirit of good old-fashioned reconnaissance, I am reporting back with my impressions of modern tech companies in general, and Google specifically.

One big surprise for me was that I got my free time back again. For reasons I fail to comprehend, I put in longer hours when I was getting paid -\$30,000 at Brown than when I'm getting paid... well, *more* than -\$30,000 at Google. I certainly found that the deadlines as an undergraduate at Brown were more numerous and more stressful than those in industry. I think it's different here because we all realize, in the back of our heads, that what we're doing must be sustainable (and must be done without long summer vacations!). The other big surprise was that I really missed TAing, as well as the relationships I had with professors and other students. This may be unique to me, but I don't find the abstract gratification of working on something that's used by hundreds of millions of people every day nearly as meaningful as dealing with "real people" one-on-one. Of course, there's lots of intra-office communication, but .com employment isn't social work. I'm sure we all knew that going in, but I didn't take the time to assess how important that was for me.

Within the domain of tech companies, though, I do believe that Google is unusual. I have the security of working at a big

company, but I also have the leverage to do what I choose: I wanted Google to show something meaningful for weather-related queries, and so I implemented that feature in my 20% time (try searching for "providence weather"); A few months ago I suggested that we give a \$5,000 incentive to all employees who purchase fuel-efficient vehicles (greater than 45 mpg), and now we do. It's nice to have that kind of autonomy. There's a [related] scarcity of managers, so engineers get to make most of their own decisions.

Google tends to hire people who are a lot like good Brown students: smart, certainly, but also well-rounded, communicative and creative. (That, in fact, is the thing I noticed first when I interviewed here, and the main reason why I chose to work at Google.) Will we retain these essential characteristics as our revenue curve becomes sub-exponential and we scale towards 10,000 employees? Who can say, but we've made it this far.

I apologize if this sounds like a sales pitch. Here's the honest truth, at least for me: I'm not sure I'll want to work at a tech company after this stint is over, but – if I do – I hope it's at a place like Google.

Best wishes to everyone at 115 Waterman,
– Special agent bhsigelm



Above: Special agent Ben Sigelman in an open space at Google hq.

Below: One of the stranger architectural features of said hq.



Teach For America

Jen Rosenbaum, '04
Castlemont High School

What with a summer teaching Providence girls how to use computers through the Artemis program and a summer at Microsoft, the Brown Computer Science Department gave me many diverse chances to explore the opportunities available to me. Yet at the end of my four years I was still unsure how I wanted to apply my degree. While the tech opportunities were increasing, my experience with Artemis piqued my interest in less direct applications of computer science.

Teach For America approached me and bombarded me with statistics: by third grade children in low-income communities are three grade levels behind their peers in affluent communities and seven times less likely to graduate from college; 50% of African-Americans do not graduate from high school; etc. Teach For America is a two-year program that places graduates from around the country in the most needy rural and urban school districts in the country. Their theory is that people who are young, energetic and knowledgeable in their field are ideal teacher candidates and that there is in general, not enough recruiting of teachers. I battled with their naiveté – how can recent graduates with no teaching experience go into the most difficult teaching situations and be expected not only to survive but thrive? But I took the challenge.

I'm now teaching high school math in Oakland and getting a new perspective on the world. It's incredible to go from Brown to Castlemont High School. Out of about 2,000 students, 100 are expected to graduate, though that number will probably go down to about 15 next year when students will be required to pass a California High School Exit Exam that tests them for graduation on 7th- and 8th-grade standards. In the entire Oakland Unified School District, only one out of 20 students complete the minimal academic requirements to attend a UC school.

I love the teaching, the students, and the chaos. I never know what students are going to do or say. I never know if it's the day when Sione, Michael and Ivette are going to puke in fourth period or if it's the day when Kenitra will finally understand fractions and how they fit into her life goal of being a chef.

I set up a computer lab with about 30 machines that my students use regularly (both academically and at lunch time to play games). My fears about taking the spot of a more qualified teacher have been assuaged by the undeniable dearth of teachers. In my small school alone (a large school that was split into three small academies), one of the three math teacher positions is unfilled, as well as two of the four English teaching positions.

I am still unsure what I want to do. But my Brown CS degree has set me up with the skills I need to handle the challenge of teaching inner-city youth and, I hope, providing them with the basic skills they need to have as many options as I do.



Glenn Carroll, Ph.D. '95 talks about his work at Google during a recent visit to campus.

Appreciative of the Time and Effort

Daniel Bilar, '95
Colby College

Alas, I was unable to attend the 25th anniversary of the department, but I was able to catch a glimpse of in the latest 'conduit' which I much appreciated. I have such fond memories of Brown CS (20 hours a week a course, the camaraderie, the 4am donuts ;)!

A little bit about me: After Brown and a one-year break, I attended Cornell University (MEng, Operations Research and Industrial Engineering) and Dartmouth College (PhD, Engineering Sciences). My field is network security. I was part of the founding group of the Institute for Security and Technology Studies at Dartmouth College. ISTS conducts counter-terrorism technology research, development, and assessment for the Department of Homeland Security. Dartmouth College filed a patent for my PhD thesis work ("Quantitative Risk Analysis of Computer Networks", advisor, Prof. G. Cybenko), which addresses risk opacity of software on wired and wireless computer networks.

I am currently a visiting professor of computer science at Colby College in Waterville, ME. I previously taught at Oberlin College. I would like to emphasize that I am successful at what I do and happy with my vocation. This is due in large part to the excellent rigorous training I got in CS11 and subsequent theoretical and applied courses. I would like to especially mention Andy van Dam, Roberto Tamassia, Franco Preparata, Leslie Kaelbling and Philip Klein in this respect. As I prepare my own courses, I see the inordinate amount of effort that goes into making lectures alluring, deep and coherent ... and I appreciate the time and effort you guys put in for me and others even more.

All the best and take care,
Daniel

CS Alums Return to Campus for Spring 2005 Career Fair



Ginger Gloystein '04 is a Program Manager at Microsoft.



Ryan Tierney '04 represents Network Appliance.



Over 1,000 students attended the spring career fair held in Sayles Hall.



Maggie Benthall '04 and Justin Boyan (a.k.a. Mr. Amy Greenwald) were busy reconnecting with old friends and answering questions about ITA Software.

ALUMNI/AE EVENTS AND HAPPENINGS

Reunion and Networking Reception

Saturday, May 28th, 5-7pm
4th floor atrium, CIT

Join computer science faculty, alums and friends for a reunion and networking reception. Enjoy good music, delicious food and drink, and stimulating conversation while touring the beautifully redesigned CIT atrium. R.S.V.P. at:

<http://www.cs.brown.edu/events/reunion/>

See you in May!

Greetings from



Center for Information Technology (CIT)

Two CS Discussion Forums Created

Two on-line discussion groups have been created to help CS alums stay in touch with each other and the department.

The first is called "Lubrano" and is a moderated e-mail announcement list for alums of Brown University's Computer Science Department and Computer Engineering graduates promoting employment opportunities, professional development and continuing education offerings, current research findings and departmental news.

Information posted to this moderated forum originates from the Department of Computer Science and from alumni/ae who contribute notices of interest to others. For more information, or to join, visit:

<http://groups.yahoo.com/group/Lubrano/>.

The second group is "CS Atrium." This will be an unmoderated networking and discussion forum for alums of Brown University's Computer Science Department and Computer Engineering graduates promoting professional and personal communication among alumni/ae. For more information, visit:

<http://groups.yahoo.com/group/CSAtrium/>.

As we all know, on-line groups are what we make of them. If you have information you'd like to share about employment opportunities, technical questions you'd like to pose, or personal news you'd like to share, please consider joining one, or both, of these groups!

Attention all CS Alums and Current Students!

Conduit! is looking for your stories, research, news and photographs for our October, 2005 issue. If you have ideas, suggestions, or would like to contribute, please contact Laura Zurowski at lpz@cs.brown.edu



Directions to Lubrano were inked on the plastic sheeting on the 4th floor.



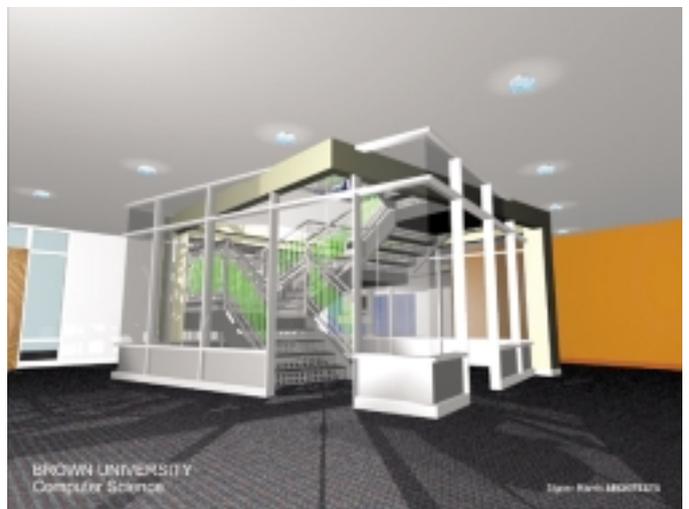
Hidden behind more plastic, the new stairs connecting 3rd and 4th floors.



C
O
N
S
T
R
U
C
T
I
O
N
M
E
M
O
R
I
E
S



More plastic sheeting covered the open area between floors. The trees were not happy!



Above: What the 3rd floor stairway will look like when completed.

Left: What the 3rd floor looked like during construction!

**A BIG THANK YOU! TO EVERYONE WHO WORKED ON,
AND WORKED THROUGH, THIS MONUMENTAL PROJECT.
WITHOUT YOU, IT WOULDN'T HAVE BEEN POSSIBLE.**

The Tradition Continues

During their stay in Greece for the Christmas break, some Greek students (all Kanellakis fellowship recipients) visited Paris Kanellakis's parents, General Lefteris and Mrs. Argyroula Kanellakis. Christos Kapoutsis, a Kanellakis fellow from MIT, joined them. It's been nine years since the tragic loss of the beloved members of the Brown CS family, Paris, his wife Maria Teresa and their children, Alexandra and Stephanos.

Manos Renieris, the first Kanellakis fellow, was the first to pay the Kanellakises a holiday visit, a visit that is now a tradition. All the fellows have developed a strong relationship with Mr. Lefteris and Mrs. Argyroula and visit them during every trip to Greece.

"We always listen carefully to Mrs. Argyroula's advice about studies and we are always impressed by Mr. Lefteris' extensive knowledge of history," say the fellows. Among many things, they discuss news of the department, Greek politics and history. "Paris is always present through the numerous photographs around the house, reminding us of his loss; but in the end," all agree, "we leave the house with a big smile."



Photo by Ioannis Tsochantaridis

General and Mrs. Kanellakis with (from left to right): Ioannis Vergados, Olga Papaemmanouil, Nikos Triandopoulos, Aris Anagnostopoulos, Christos Kapoutsis (MIT)

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

Non-Profit
Organization
US Postage
PAID
Permit No. 202
Providence, RI