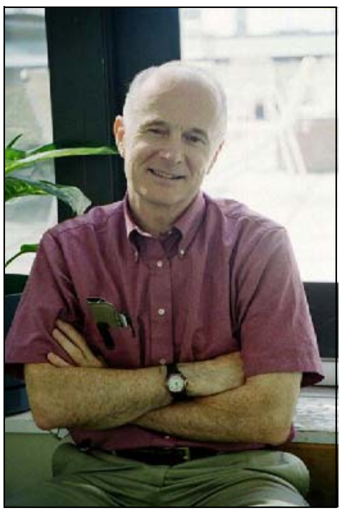
conduit!

Volume 12, Number 1 | Department of Computer Science, Brown University | Spring, 2003

COMPUTATIONAL NANOELECTRONICS



John Savage

The Integrated Circuit

Moore's Law postulates that the number of transistors in an integrated circuit (a chip) will double every few years. Remarkably, this forecast of exponential growth has been accurate for more than 45 years. The result has been a rapid decline in the cost of computers and a rapid increase in their speeds and storage capacities. This technological revolution has led to productivity gains of historic proportions.

Moore's Law is a consequence of continuing advances in photolithography, the process of inscribing transistors and wires on the surface of crystalline semiconductor materials. Photolithography is a multi-step process in which light passes through masks whose openings specify the location of wires and devices and falls onto photosensitive coatings on a semiconductor wafer. Solvents then remove the areas exposed to light so that they can be used for wires and devices.

Today photolithography is complex and precise. Features as small as 90 nanometers (10^{-7} meters) can now be imprinted on chips. This sophistication is costly; the ultraclean factories required to manufacture such high-density chips cost several billion dollars to construct. Unfortunately, it is predicted that the exponential growth in chip capacity through photolithography will end within at most ten to fifteen years, the normal length of time for research to reach the marketplace. The resolution achievable through photolithography is determined by the wavelength of electromagnetic radiation used, and we are already operating at high frequencies and small wavelengths.

The Future Lies with Nanotechnology

Fortunately, nanotechnologies appear to have the potential to supplement photolithography and allow chips of increasing density. So what is nanotechnology? Nanotechnology is the manufacture of technologies whose smallest dimension is on the order of a few nanometers. Chemists have been able to grow carbon nanotubes (CNTs), tubes made of carbon atoms, and semiconducting nano-wires (NWs), crystalline wires made of a semiconductor material, both of which make good wires. Diodes and transistors can also be formed from CNTs and NWs. Thus, the building blocks for computers can be realized from nanomaterials. Nanotechnology chip manufacture is expected to use **self-assembly**, a process in which very large numbers of objects with nanometer dimensions arrange themselves into stable configurations. Chemists are adept at manufacturing molecular structures using self-assembly. However, since this process is primarily stochastic in nature, while nanotechnology chips are very likely to exhibit a high degree of regularity, the exact location of individual objects, such as CNTs and NWs, probably cannot be controlled.

Today photolithography is complex and precise. Features as small as 90 nanometers (10^{-7} meters) can now be imprinted on chips

Brown University, Box 1910, Providence, RI 02912, USA

Crossbars, An Important Architectural Element

So we ask, what types of computers are likely to be built using nanotechnologies? The current thinking is that crossbars, a two-dimensional array (**nanoarray**) formed by the intersection of orthogonal sets of parallel and uniformly spaced NWs organized into rows and columns, will be an important technology (see Figure 1a).

Researchers have shown that nanowires can be aligned in parallel with nanometer spacings using directed self-assembly and that two such layers of nanowires can be assembled at right angles. They suspend NWs, which are so small as to be invisible by a light microscope, in a solution and float them through a trough on a chip. The NWs orient themselves as would logs in a stream and come to rest aligned in the direction of the flow. Materials have also been developed that permit nanoarrays to store binary data at a **crosspoint**, the intersection of a pair of orthogonal nanowires. Thus, nanoarrays for data storage are a realistic possibility.

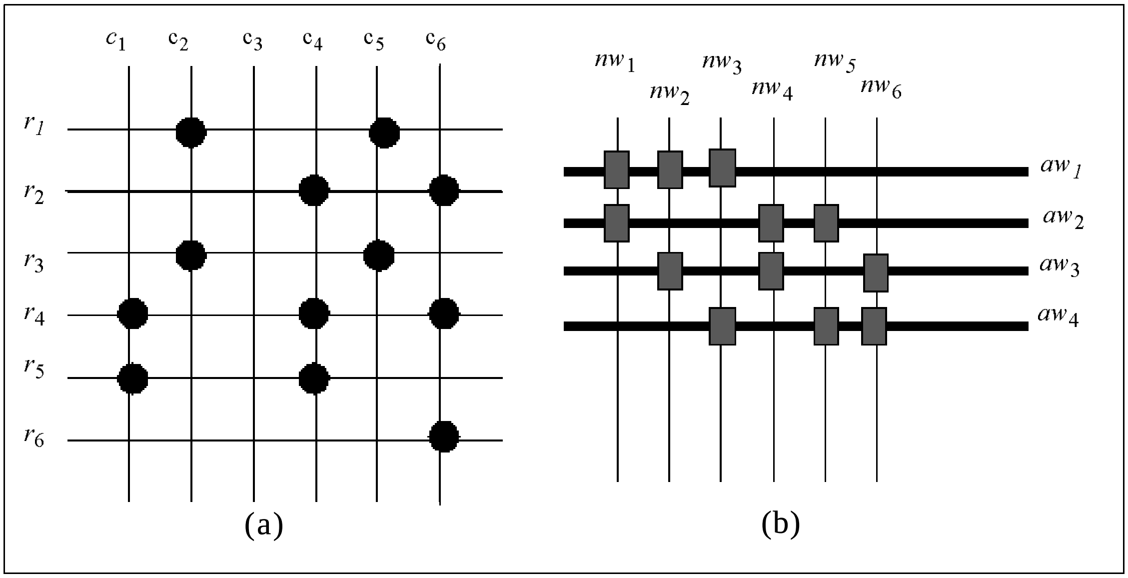


Figure 1: (a) A nanoarray with dots showing locations of 1s; 0s are in the other locations. (b) 2-hot addressing of six nanowires $\{nw_1,...,nw_6\}$ with four microlevel address wires $\{aw_1,...,aw_4\}$. In practice the microwires would be much larger than the nanowires

Addressing Nanoarrays

Access to nanoscale devices will necessarily be from the micro level. If nanowires must be attached individually to microwires, no advantage will accrue to nanotechnology. Thus, to use nanoarrays efficiently some method must be found to address a large number of NWs using a small number of microwires. André DeHon of Caltech, Patrick Lincoln of SRI and I have proposed a way to construct an address decoder solving this problem that uses the fact that *p*-type regions can be doped into NWs along their length. Each region is conducting unless a sufficiently large positive electric field (denoted by 1) is applied above. Thus, a wire placed above or below a region at right angles forms a field-effect transistor (FET) that acts like a switch and is turned off when a sufficiently large positive voltage is applied on an orthogonal wire. We use this result to design a decoder.

We assign to each NW a binary code, called an *h*-hot address, containing *h* 1s and $(b - h)$ 0s. We assign a 1 to an undoped region and a 0 to a doped region. We propose that coded NWs be laid down at right angles and on top of a set of microwires that overlap the doped and undoped regions, as suggested in Figure 1b. If all possible codes appear among the NWs, applying a high voltage to *h* of *b* microwires will result in only one NW that is conducting. This allows each NW to be addressed.

Given the stochastic nature of directed self-assembly, it cannot be ensured that every differently coded NW will appear. Similarly, NWs are unlikely to align perfectly with an orthogonal set of microwires. We have shown how to cope with these uncertainties to design a decoder that can be used in a nanoarray. Ours is the second decoder for NWs to be proposed; the first, based on the random deposition of gold nanoparticles on the intersections of the NWs and microwires, has its own set of manufacturing challenges. Our decoder provides an important technique for bridging the gap between microscale and nanoscale features and for bootstrapping the programming and customization of nanoscale systems.

Storing Data in Nanoarrays

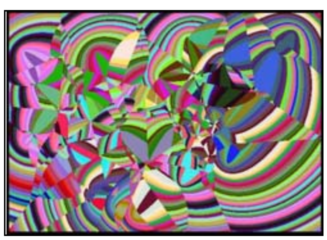
Nanoarrays can be used for data storage. Researchers have developed molecular films that, when placed between the row and column NWs of a nanoarray, become conducting (nonconducting) at the intersection of rows and columns when a large positive (negative) voltage is applied between these rows and columns. Thus, if a large positive (negative) voltage is applied across a set *R* of rows and a set *C* of columns, the film in the regions around the crosspoints formed by these rows and columns becomes conducting, represented by 1s (nonconducting, represented by 0s). This property means that nanoarrays should be able to store large amounts of data. We say that a *store* (*restore*) operation is done when 1s (0s) are written. Nanoarrays can also be used to compute, since one row or column can be seen as computing the "wired OR" of its input. Since inverters can be constructed using NWs, every Boolean function can be realized by these nanoarrays.

Lee-Ad Gottlieb, Arkady Yerukhimovich and I have considered the following questions: a) What are the most efficient ways of entering data into large nanoarrays? b) How difficult is it to find a minimal or near-minimal number of stores and restores to enter the data?

We have shown that the answer to the second question when the normal data entry method is used, i.e. when only stores are performed (restores are not used), is that it is NP-hard. In addition, we have shown that it remains NP-hard to obtain good approximations to the minimal number of operations to program a nanoarray. However, when a fixed upper bound is placed on the number of 1s in rows or columns, although the program remains NP-hard, reasonably good approximations can be obtained in polynomial time. If only stores were available to program nanoarrays, we might conclude from these results that the potential of nanoarrays is indeed limited. Fortunately, for some prototype problems nanoarrays can be programmed much more efficiently with stores and restores than with just stores alone. For example, an $n \times n$ nanoarray with 1s on the diagonal requires n stores but only $2\log_2 n$ stores and restores. Although this suggests that the problem of minimizing the number of operations might not be NP-hard, we have developed a complex proof that the problem remains NP-hard.

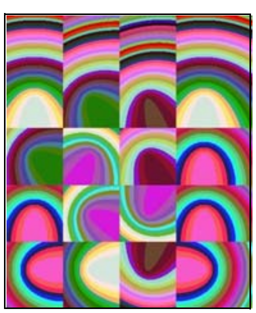
The above results apply when each row and column NW can be individually addressed. Since nanoarrays accessed from the micro-level will probably use a form of *h*-hot addressing, we have also examined the same questions when *h*-hot addressing is used. While individual nanowires can be controlled via decoders, decoders also constrain the set of wires that can be addressed simultaneously. As a consequence, more time will be needed to enter data into a nanoarray in this case. Because of the diffculty of programming nanoarrays with *h*-hot addressing, we have explored the programming of individual rows or columns. We have shown that it is NP-hard to find a minimal or near-minimal number of steps to program a single row or column of an array. However, we have also shown that this problem is log-approximable when *h* and *b* are proportional to the logarithm of the number of

HAMILTONIAN LANDSCAPES



These beautiful images were created by Selim Tezel, M.Sc. '02, using the software he designed for his Master's project, "Traveling Salesman's Sketchpad (TSSP 1.0)." The image to the left was a finalist at **Manifest 2003**, a juried digital visual arts festival, part of this spring's international Boston Cyberarts Festival.

Designed and implemented to study the Euclidean Traveling Salesman Problem, the process starts by placing a number of nodes (cities) on the plane that the salesman is to travel. Then an algorithm is chosen by the user (nearest neighbor, cheapest insertion, etc.) to compute an approximately optimal tour (approximate since the problem is NP-complete, i.e. not generally solvable in real time). The images rendered are isoperimetric and isopermutation regions associated with the chosen algorithms, demonstrating the constructed tours' sensitivity to initial conditions. A full description of the process with a glorious image gallery and supporting documents is at:
http://www.cs.brown.edu/people/stezel/webPageDesign/thesis.html

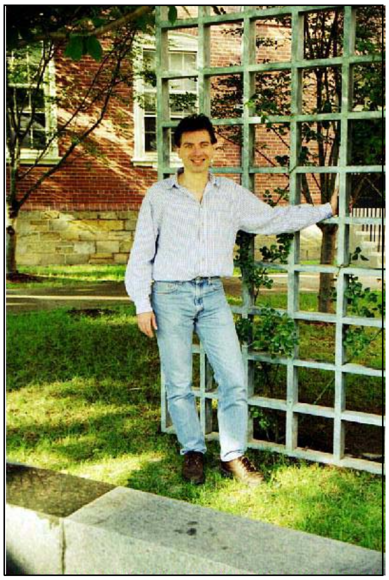


rows and columns in an array. Finally, we have shown that highly efficient algorithms exist for programming some prototypical arrays when *h*-hot addressing is used.

Conclusion

Nanotechnology offers many interesting challenges. Although the building blocks for nanocomputers have not been fully developed, there is a role here to be played by computer scientists in shaping the development of this emerging field.

COMET, OR HOW TO FIND THE RIGHT NEIGHBOR



Pascal Van Hentenryck

Introduction

Far from the public eye, modern societies often rely on optimization technology in order to function effectively. From the airline industry to courier services, from supply-chain management to manufacturing, from facility location to resource allocation, many important decisions are taken by optimization software every day. Even the entertainment industry is no exception: sport leagues, which are significant sources of revenue for radio and television networks around the world, generate extremely challenging scheduling problems.

Most optimization problems are challenging from both a computational and software engineering standpoint. Generally, they cannot be solved in polynomial time and require expertise in algorithmics, applied mathematics, and the application domain. Moreover, the resulting software is often large and intricate, which makes it complicated to design, implement, and maintain. For instance, the apparently simple task of dispatching technicians to repair telephone lines requires about 150,000 lines of complex C++ code. Since many optimization problems can be specified very concisely, this is very ironic. The distance between the specification and the final program is thus considerable, which indicates that software tools are seriously deficient in expressiveness and abstractions in this application area.

Because of the nature of optimization problems, no single approach is likely to be effective on all of them or even on all instances of a single problem. It is thus of primary importance that all major approaches to optimization be supported by high-level tools automating many of the tedious and complex aspects. Historically, most research has focused on constraint and mathematical programming, and these approaches are now supported by a rich variety of modeling and programming tools. In contrast, neighborhood search, one of the oldest optimization techniques, has been largely ignored until recently. This is a serious gap in the repertoire of optimization tools, a limitation further exacerbated by the fact that solving optimization problems remains a highly ex-perimental endeavor: what will or will not work in practice is hard to predict. Proper software tools facilitate this experimentation and often yield higher-quality solution techniques, since users are more likely to try out various avenues.

At a conceptual level, neighborhood search explores a graph whose nodes represent solutions (or configurations) and arcs represent a transition from a solution to a neighboring solution. How to define this neighborhood graph and how to explore it effectively are fundamental issues that have recently received considerable attention.

> *The distance between the specification and the final program is considerable, showing a serious lack in expressiveness*

Neighborhood search is the technique of choice for a variety of fundamental applications. For instance, at the time of writing, the best approach to the traveling tournament problem, an abstraction of major-league baseball scheduling, is a neighborhood-search method that significantly outperforms constraint and mathematical programming. The same is true of many important problems, such as vehicle routing, frequency allocation, and many resource allocation and scheduling problems. Equally important perhaps is the belief that hybrid algorithms that combine several approaches in innovative ways are likely to produce the next improvements in this area. Recent results in routing and scheduling indicate the promise of hybridization. Yet neighborhood-search algorithms are weakly supported in modeling and programming tools.

The COMET project was initiated to address these needs. COMET, an object-oriented language supporting a constraint-based architecture for neighborhood search, features novel declarative and control abstractions. It decreases the size of neighborhood-search programs significantly and enhances compositionaly, modularity, and reuse for this class of applications.

The COMET Architecture

The neighborhood-search architecture supported by COMET (Figure 1) consists of a declarative and a search component organized in three layers. The kernel of the architecture is the concept of *invariants* over algebraic, set, and graph expressions. Invariants are expressed in terms of incremental variables and specify a relation that must be maintained under assignments of new values to its variables. For instance, the code fragment

```
inc{int} s(m) <- sum(i in 1..10)  a[i];
```

declares an incremental variable `s` of type `int` (in a solver `m`) and an invariant specifying that `s` is always the summation of `a[1],...,a[10]`. Each time a new value is assigned to an element `a[i]`, the value of `s` is updated accordingly (in constant time). Note that the invariant specifies the relation to be maintained incrementally, not how to update it.

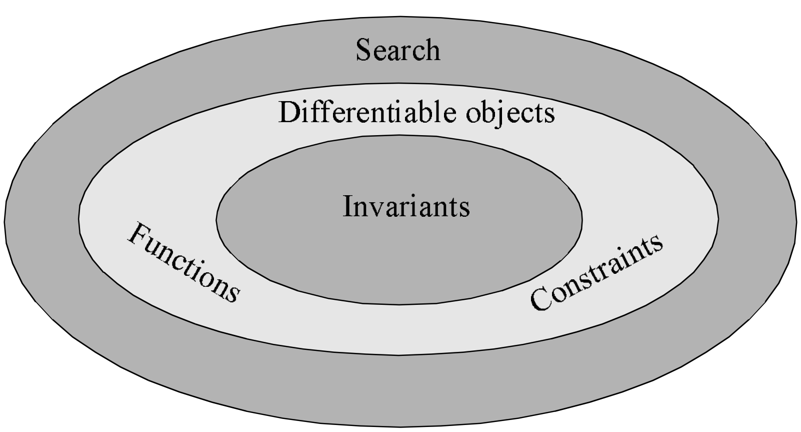


Fig. 1. COMET's constraint-based architecture

Once invariants are available, it becomes natural to support the concept of *differentiable objects*, a fundamental abstraction for neighborhood search. *Differentiable objects maintain a number of properties (using invariants or special-purpose incremental algorithms) and can be queried to evaluate the effect of local moves on these properties.* They are fundamental because many neighborhood-search algorithms evaluate the effect of various moves before selecting the neighbor to visit. Two important classes of differentiable objects are constraints and functions. Constraints are a natural vehicle to model combinatorial optimization problems, as past research has shown, but play a fundamentally novel role in this architecture. Instead of pruning the search space as in systematic search algorithms, constraints incrementally maintain properties, such as their violation degree, and can be queried to evaluate the effect of transitions on these properties.

These first two layers, invariants and differentiable objects, constitute the declarative component of the architecture. The third layer of the architecture is the search component, which aims at simplifying graph exploration. Neighborhood-search algorithms often induce intricate control flow, interleaving various application aspects such as the neighborhood exploration, the heuristic, and the metaheuristic. The control abstractions of COMET aim at separating these orthogonal features in the source code, enhancing compositionality, modularity, and reuse of the search components. Events and first-class closures connect these features conveniently by separating dynamic behaviors from their uses and when they must be used.

Figure 2 depicts a simple COMET program to solve the queens problem, placing n queens on an $n \times n$ chessboard so that no two queens attack each other. Line 5 declares an array of incremental variables and `queen[i]` specifies the row of the queen in column `i`. Lines 8-14 specify the declarative component, i.e., the constraints that must be satisfied by a solution. The constraints in lines 9-11 specify that the queens cannot be on the same row and the same diagonals. They use the ubiquitous `Alldifferent` constraint, a differential object that arises in numerous resource-allocation applications. All the constraints are stored in a constraint system (also a constraint) that can be then queried in various ways. Constraint systems are one

```
1.  int n = 512;
2.  range Size = 1..n;
3.  LocalSolver m();
4.  UniformDistribution distr(Size);

5.  inc{int} queen[i in Size] (m,Size) :=distr.get();
6.  int neg[i in Size] = -i;
7.  int pos[i in Size] = i;

8.  ConstraintSystem S(m);
9.  S.post(new AllDifferent(queen));
10. S.post(new AllDifferent(queen,neg));
11. S.post(new AllDifferent(queen,pos));
12. inc{set{int}} conflictSet(m);
13. conflictSet <- argMax(q in Size) S.violations(queen[q]);
14. m.close();

15. while (S.violationDegree())
16.   select(q in conflictSet)
17.      selectMin(v in Size)(S.getAssignDelta(queen[q],v))
18.         queen[q] :=v;
```

Fig. 2. The queens problem in COMET

of the main tools to enhance compositionality and make it easy to add new constraints without changing the rest of the program. Line 13 is particularly interesting: it specifies `conflictSet`, the set of queens under attack. This set is specified by querying the constraint system and is automatically maintained by the COMET implementation whenever queens are moved. Lines 15-18 specify the search, which consists of choosing a most violated queen and moving it to a new position in order to minimize its violations. Observe the disconnection between the problem constraints and the search. Whenever line 18 is executed, all the constraints and invariants are automatically updated by COMET, although they are physically separated in the source code.

It is interesting to discuss enhancing this program for algorithm animation. Assume that we have at our disposal an `Animation` class that provides a method

`updatePosition(int q,int p)`

to update the row of the queen on column `q` to `r`. It suffices to add

```
forall(q in Size)
  whenever queen [q]@changes(int or,int nr)
    animation.updateQueen(q,nr);
```

just between lines 14 and 15. The code features an event `whenever` that executes the enclosed closure whenever a queen is moved. Once again, there is a complete textual separation between the animation code and the rest of the application. Note that solving the 2K queens problem takes about one second on a modern PC.

The above example is of course very simple. What is interesting is that larger programs share the same structure and simplicity. For instance, a job-shop scheduling program written by Keith Schmidt (ScM '01) about 5,000 lines of C++ long now takes less than 500 lines of COMET for the same runtime efficiency.

Implementation and Future Development

The COMET system is being developed jointly with Laurent Michel (PhD '99), now an assistant professor at the University of Connecticut. It consists of an interpreter, a JIT compiler, a runtime system, and many libraries. The overall system is now close to 200,000 lines of C++. The core, i.e., the compiler and runtime system, is about 50,000 lines long, the libraries now becoming the major part of the system. COMET has been evaluated on many applications. It has been shown to be competitive with special-purpose C++ programs on a variety of scheduling and resource allocation problems, while reducing the size of the code significantly. More importantly, perhaps, COMET was key in helping us develop new neighborhood-search algorithms in car sequencing, warehouse location, and scheduling that have solved open problems and/or outperformed existing algorithms.

Given the speed of progress in neighborhood search, COMET is only a first step in a fascinating area. Numerous open issues remain, ranging from language design to runtime support and incremental algorithms necessary to accommodate a wide variety of new results (e.g., very large-scale neighborhoods and scatter search). Paramount among the language-design issues is the need to preserve the compositionality, modularity, and efficiency of COMET for applications with numerous side-constraints gravitating around the core structure. Practical applications naturally exhibit these "exotic" constraints: just think about all the safety regulations in the airline industry or, on a lighter note, the fact that Duke must meet UNC on weeks 11 and 18 during the NCAA tournament, destroying the symmetry of the round-robin tournament. And we are not even talking about the constraints imposed by the NFL!

HOW A SMART ELEVATOR CHANGED OUR WAY OF LIFE FOREVER

Guy Eddon (l) and Matt Lease

It seems hard to imagine now, but there was a time before all the Gap ads, before the appearances on Oprah and the late-night talk shows, before the piles of fan mail and invitations to posh parties had begun to arrive, a time before anyone knew we were anything but a couple of graduate students with a crazy idea. No one believes us now when we try to explain how something so simple revolutionized our way of life so completely, but the world needs to understand how it all really happened. At the time they told us it was impossible, that nothing like what we envisioned had ever been done or could be done, that we should put our youthful dreams aside and face up to the reality of what could never be. But like so many other dreamers, we persevered in spite of their criticism and the other challenges we met, for we knew if we could pull it off, if we could do this thing they called impossible, it would be like nothing the world had seen before.

But I am getting ahead of myself now, for I wanted to tell you how it all began. We were both young then, new students in the Ph.D. program, in fact. Guy Eddon, my partner in crime, was a struggling writer who had made ends meet over the years writing Harlequin novels while secretly pining for a better life. He hoped to find it in a town called Providence. For my part, my dot-com days of selling fake vomit over the Web had gone bust like so many other tinsel dreams of the web craze. I thought maybe I could make it as a pimp in Frank Woods' escort service, but when business went dry, I decided to give Computer Science a try. In short, Guy and I were a couple of down-and-outers looking to better our fate.

We had no reason to end up any better than all the other poor saps who entered the program, but we caught a break early on when one of the greats of those days, Ugur Cetintemel, took us under his wing. Ugur began to teach us of the black arts he had devoted his life to mastering: Pervasive Computing, sometimes also called Ubiquitous Computing. It was then that we caught our first glimpse of this netherworld, a place where the convergence of Moore's Law and the vision of Mark Weiser made it possible to imagine embedding computation and communication into virtually any manufactured object. In this world, the ubiquitous presence of sensors and actuators enabled computers to maintain a constant vigil monitoring human needs and the state of the world, continually seeking new opportunities to support human activities through appropriate physical and digital intervention. But Grand Master Cetintemel had one problem he hadn't been able to solve: how could this netherworld and the world we knew be brought together? How could all of the many interesting everyday artifacts and legacy devices from our world be enabled to participate in this digital netherworld when they lacked the requisite computational abilities?

It was then that the idea took seed in us and began to grow. Legos had solved all of our problems when we were boys; could they save the day once again? It seemed so obvious to us: Lego Mindstorms were cheap and could sense, actuate, compute, and communicate—why not use them to sense and control everyday devices and revitalize them as new, pervasive artifacts? When we shared our idea with our friends and colleagues, we were universally condemned, belittled, and scoffed at, but who could blame them? You must remember that no one knew about the

no one dreamed that a simple child's toy could hold the key to such a breadth of problems as NP-completeness, world hunger, and the six degrees of Kevin Bacon

Universal Turing-Lego Thesis then. Back in those days "real" research required math, simulation, and independent thinking: no one dreamed that a simple child's toy could hold the key to such a breadth of problems as NP-completeness, world hunger, and the six degrees of Kevin Bacon. As we've said in the many interviews since then, we didn't have any idea ourselves of the revolution we were starting—all we wanted was to prove ourselves to the Grand Master and earn an A in CS295-1.

We needed a way to test our idea, however, and it was then that we turned our gaze upon the CIT's elevators. When we had first arrived at Brown, we were amazed at the irony of the school's Center for Information Technology being saddled with a legacy elevator system widely regarded as broken. Despite the close physical proximity of the two elevators, they had been installed without any coordinating logic—each had separate call buttons and ran independently of the other. Even worse, the only visible indicators of the elevator's position were in the lobby. Consequently, riders on other floors had no idea which elevator to call for quickest service or how long they might have to wait for an elevator to arrive. While we had no desire to mess around with elevator internals (we didn't care how the elevator's motors operated and certainly didn't want to do anything requiring a new safety review of the installation), we figured we could augment the elevator with Lego Mindstorms not only to solve the problems above but also to revitalize the elevator as a pervasive device. Not only would the system be remotely operable (one would be able to call an elevator in advance or see if one had to hurry to catch it), but, by combining a location-aware badge system with an electronic appointment calendar, maybe you would never have to call the elevator at all: the elevator would be able to anticipate your need for service.

Although the Grand Master had his doubts, he consulted his brethren in the Dark Arts, and Overlord Stan Zdonik saw fit to provide us with the Lego Mindstorms we needed to undertake our mission. We toiled and slaved away in the dark recesses of the CIT, breaking from our labors only long enough to partake in the weekly TGIF festivities and consume sufficient Klondike bars to fuel our efforts. We faced many trials and tribulations during the ordeal. When deploying our devices in the lobby, the evil ones, those who actually wanted to use the elevators, would see the exotic lights and wires of our devices and constantly bombard us with nefarious questions like, "Is the elevator working?" Yes, we would say, again and again, the elevator is working, but then another minion of evil would approach us with the same question, thinking such incessant distraction would prevent us from realizing our vision. But all our engineering training was not for naught, for we conceived a plan to beguile our assailants: we posted signs with craftily designed messages like "THE ELEVATOR IS WORKING!!" And when they thought to ask us, "What are you guys doing anyway?", we handed them a flier and resumed our task. El Cid would have been proud of our determination.

But the forces of evil were not to be assuaged so easily. One of the truly evil ones—you know, those who get off at the third floor—summoned the night watch to harass us further: "It's been reported that you guys are hacking the elevator control system in order to bypass security protecting the basement!" To overcome assaults, we obtained a scroll from Grand Master Cetintemel testifying to the righteousness of our endeavor and threatening his great wrath upon any who opposed us. With scroll in hand, our opponents quivered in fear and none could stand against us. Our toil continued, day after day. There did come a time when we doubted, when it seemed we lacked sufficient Legos to accomplish our mission, but our prayers were answered when his exalted mightiness Eugene Charniak saw fit to bestow upon us a plethora of Mindstorms not to be used for the Robots course until the following semester. And so we resumed our labors once more.

At last the day of presenting arrived. We plugged in all the wires, added the batteries, and attached the most powerful device of all, actual duct tape, in great quantity. It worked, and all who saw what we had accomplished marveled at its splendor. It was remarked that, after 15 years of no progress in fixing the elevator, a couple of graduate students had come along and shown it was simple enough to be solved using Legos. The faculty swarmed around our invention, discussing how all of the old textbooks needed to be rewritten, courses redesigned, and accepted practices revisited, and visitors from all the lands came to see what we had done. Only then did Guy smile, for he knew then that his days of Harlequin writing were behind him and a bright new future was unfolding before us.

Looking back now, who could have guessed that would only be the beginning? It really wasn't until the Universal Turing-Lego Thesis had been proven that we really began to understand how deeply our work had changed the world, that nothing would ever be the same. For us, life has been a confused whirlwind of book signings and Turing awards, extensions to Mount Rushmore, parties with Yanni, etc. I've come to realize that being famous, rich, and powerful is not always easy—I get hand cramps now and then from answering a mere fraction of the fan mail—but I sleep better at night knowing we've made the world a better place, free from fear and envy and the injustices of the past, a place where universities understand the critical role of Legos in any serious research agenda.

"SmartElevator: Revitalizing A Legacy Device Through Inexpensive Augmentation" will be presented at the 3rd International Workshop on Smart Appliances at ICDCS'03 in Providence, May 19th. Tickets should be purchased in advance as the event will, of course, be standing room only.

Technical staffer Max Salvas adding 256MB of memory to Suzi's Maxbuilt *system as well as upgrading her CPU from 800MHz to 1.2GHz. These enhancements were necessary to support the performance needed to produce this issue of **conduit!***

During the past year the department replaced most of its desktops with Maxbuilts. *During the summer of 2002, Max and Charles Williamson, the Student Hardware Technician, built more than 250 of these systems, bringing the department total to over 360. In addition to the generic* Maxbuilt *(1.5GHz CPU, 500MB memory, 40GB disk space, and GeForce3 graphics), Max has built several custom systems with high-performance dual processors and large amounts of memory. Max received a University Staff Bonus Award for his efforts and successes in the recent upgrade project.*

THE SECOND KANELLAKIS MEMORIAL LECTURE

Christos Papadimitriou surrounded by Kanellakis Fellows. l to r: Yiannis Tsochantaridis (Brown), Christos Kapoutsis (MIT), Manolis Kamvysselis (MIT), Nikos Triandopoulos (Brown), Manos Renieris (Brown), Olga Papaemmanouil (Brown), Aris Anagnostopoulos (Brown), Yiannis Vergados (Brown)

The second annual Kanellakis Memorial Lecture was presented last December 9 by Prof. Christos Papadimitriou of the University of California, Berkeley. In memory of our colleague Paris Kanellakis, who died with his family in an airplane crash in December 1995, these lectures are held around Paris's birthdate, December 3.

Papadimitriou spoke on "Algorithmic Problems in the Internet," discussing how the Internet, and particularly the World Wide Web, has influenced research in theoretical computer science. Christos expressed his belief that Paris, a person of broad intellectual pursuits who loved to tackle hard problems, would have enjoyed working on the sort of problems detailed in his talk.

We were especially pleased that in addition to the past and present Brown recipients of Kanellakis Fellowships, the holders of this year's Kanellakis Fellowships at MIT were also present.

THE 30th IPP SYMPOSIUM INFORMATION & KNOWLEDGE MANAGEMENT

Thomas Hofmann

On November 14, 2002, the Department hosted the 30th IPP Symposium on "Information and Knowledge Management", a topic at the frontier of today's computer science research: How can computers support the creation, preservation, and dissemination of information and knowledge? Three key issues came up time and again in the talks and discussions. First, how can we build search engines and retrieval systems that can better deal with the ambiguities and inherent vagueness of natural language? Second, how can we extract useful information or knowledge from repositories like the World Wide Web? And third, how can we make multimedia documents such as video and audio searchable?

The meeting was kicked off by **Bill Woods of Sun Microsystems**, who addressed practical knowledge representation for efficient support of information access and information sharing within an organization. One of the fundamental problems in this context is apparent to anyone who has used information-retrieval tools such as search engines: not everybody uses the same words and phrases to express a certain fact, concept or idea, as in "calling a spade a bloody shovel". This is called the vocabulary mismatch problem. To address this problem, Bill has devised a method that uses a concept taxonomy to match a search request with a document on the concept level (not simply by comparing keywords, as most search engines do). Thus the system can understand that someone looking for information on "dogs" may be interested in a document on "German shepherds" but is less likely to want information on "underdogs".

Symposium speakers l to r:
Andrew McCallum, UMass Amherst; Bhiksha Raj, MERL; Susan Dumais, Microsoft Research; Bill Woods, Sun; Alex Cozzi, IBM Almaden, center; Pedro Moreno, Hewlett Packard; John Smith, IBM Watson; and host Thomas Hofmann

Bhiksha Raj and Peter Wolf of MERL Mitsubishi Research Lab in Cambridge presented SpokenQuery, a system that provides a spoken interface to an information-retrieval system. Ever want to google through your cell phone or PDA? Don't like the foldaway keyboards, and don't like drawing your query in glyphs either? Well, then, SpokenQuery is what you (will) need. Just speak what you want to know into a microphone! Its automatic speech recognizer is optimized to get at least the essential parts of your query right, i.e. those important keywords that really matter. It does this by coupling the speech recognizer with the document index, since it is the index that is ultimately used to retrieve relevant information.

The next presentation was given by a researcher with a stellar reputation in both the academic and commercial world, **Andrew McCallum of UMass Amherst**, former VP of R&D at WhizBang Labs. His research is driven by one key question: how can we transform the "Wild Web" into a knowledge base? To demonstrate the feasibility and the commercial value of this idea, Andrew and his coworkers at WhizBang Labs developed the FlipDog job search engine (*www.flipdog.com*). Using innovative information-extraction tools, FlipDog constantly spiders the Web in search for its prey: job ads. Once a job ad is found, relevant information such as company name, job type, department, level, location, etc. is extracted from the HTML file and mapped to a standard database record. With this technique, FlipDog has automatically collected and processed more than 200,000 job ads—and keeps on going! At its core, this impressive system is powered by sophisticated machine-learning algorithms using statistical models that improve on a standard technique known as hidden Markov models—a wonderful example of how progress in modeling and theory can make a real difference in practical applications.

Susan Dumais of Microsoft Research in Richmond, a renowned research veteran, presented research by her and her colleagues on question-answering retrieval using the World Wide Web as a knowledge source. This addresses one of the longstanding dreams of artificial intelligence: how can we build a computer that can answer any question we mere humans may ask? Well, why not use the hidden knowledge treasures of the Web to do it! Of course, this will just be a collective echo of everyone who has put content on the Web, but maybe this is sufficient for many questions people have. So how, in a nutshell, does their system, called AskMSR, work? First, the question is transformed into a number of answer templates. You ask, "Where is the Louvre Museum located?" and AskMSR looks for Web pages containing phrases like "the Louvre Museum is" and inspects the context to the right. The collected snippets are then filtered and finally tiled together based on redundancy and overlap to create the answer. Does it work, you ask? Amazingly well: "Who is Bill Gates married to? AskMSR's answer is "Microsoft".

In their talk on Model-Based Retrieval and Multimedia Information and Biosurveillance, **Chung-Sheng Li and John R. Smith** presented some recent research projects at **IBM Watson Labs** on multimedia retrieval and data mining. Among the many highlights of their talk was their account of a system for biosurveillance, a prototype of which was deployed at IBM Watson. The system aims at automatically detecting early warning signs of possible outbreaks of epidemic diseases, whether of "natural" or "less natural" causes. The system collects data on personal activities like phone calls, employees' arrival and departure times, Internet access patterns, etc. to mine for irregularities and alarming trends in people's behavior. The idea is that a system like this can be an early warning system to give precious lead time compared to traditional channels that rely on sources like medical reports.

Existing video libraries are growing rapidly. Soon we will be facing the problem of finding relevant information in the giga- or terabytes of data stored. **Alex Cozzi of IBM Almaden Research** described CueVideo, a prototype system developed at IBM that can create browsable video summaries and automatically index video footage based on an audio transcript. Say you own (or illegally copied...oops) a movie like *Pulp Fiction* and you want to find the scene on "foot massage" or the "catch-up/ketchup" joke. CueVideo might help you get right to the point. Of course, you can also use software like this to search through the webcast of your favorite CS course for keywords like "inheritance" and "polymorphism", but what would be the fun in doing that?

In the last talk of the symposium, **Pedro Moreno of Hewlett Packard Research Labs** presented a truly fascinating system for audio retrieval called SpeechBot. Collecting hundreds of gigabytes of data from radio news feeds, SpeechBot automatically transcribes the speech to text and makes the information searchable by building an index. SpeechBot (*speechbot.research.compaq.com*) currently indexes an impressive 16,000 hours of radio—that's almost three years of listening day and night! The main challenge is that the transcripts generated by speech-recognition software rarely exactly match what was spoken, because different people speak at different speeds and with different degrees of clarity. However, meaningful words are recognized with high accuracy and even when a word is missed, it will probably be recognized when it is spoken somewhere else in the program. In addition, a number of innovative techniques have been implemented to counteract the unreliability of today's automatic speech-recognition systems, for example, the use of sub-word units, so-called 'particles', to index spoken documents. For example, the word "pescatarian" is unlikely to be recognized by any speech-recognition system, but the particle "tarian" may be useful as an index particle that can be matched with queries like "vegetarian".

Now let's hope all this will improve the state of the art in intelligent information retrieval. If you AskJeeves, "The meaning of life is to increase fitness". Am I missing something here?

Dr. Tom Tullis, Sr. VP of Human Interface Design at the Fidelity Center for Applied Technology, joined Michael Black's CS32 Intro. to Software Engineering class to lecture on "Effective User Interface Design". Tom founded and runs the Usability Lab in Boston.

Fidelity Investments is a CS Industrial Partner.

changelog

*Andy was delighted to report that his student **Leah Pearlman**, a junior and Sc.B candidate has been awarded a Microsoft Scholarship for next year. This scholarship covers full tuition and includes a 12-week salaried summer internship at Microsoft. Our thanks to Microsoft and congratulations to Leah!*

------8<------

*Doctoral student **Manos Renieris** received rather back-handed kudos as a result of his failed application to the ICSE '03 Doctoral Symposium. Said his reviewer, "Your application was reviewed as something perhaps best described as 'Has no real need of the Symposium.' The application showed that you are making excellent progress under excellent supervision, and with considerable success in the field already. However, we want you to know that except for being*

LETTERS FROM ALUMNI AND FRIENDS

ED BIELAWA, ScB & ScM '95.5

After eight years and six companies riding the Internet startup wave (with one buyout but otherwise a lot of stock certificate "wallpaper"), I'm currently working on the other side of software development: I'm a product manager at Phonetic Systems in Bedford, MA, where I define the requirements of our speech-recognition platform and manage its integration with partners such as Oracle, AT&T, and IBM. My focus is twofold: on the enterprise side, to increase our market penetration by partnering with interactive voice-response systems (Brown is using our system to voice-enable the campus directory), and on the telco side, to build and deploy the first fully automated national directory assistance "411" system.

Contrary to popular belief, directory assistance is not automated today. Even though a computer prompts callers for their listings, live operators listen to the recordings and manually enter the requested listings into a computer. While speech recognition on small databases is almost a commodity nowadays, speech recognition on the entire US white pages (150M+ listings) is a very difficult task. First, the database requires significant normalization, since listings have been entered in many different ways by many different operators over the years. For example, names and roads can be abbreviated in many different ways, and callers never ask for listings the way they appear in the phone book—just think about how many ways you can ask for the phone number of the CIT! And that's assuming callers respond to prompts in the correct format—but many callers speak as freely as though the system could understand natural language. Fortunately, the number of calls to directory assistance is somewhat limited, but we still need technology to handle free-form natural-language responses (Eugene's CS241 has come in handy!). Even after the database is normalized and callers are directed to correctly formatted responses, speech recognition on large-metro-area databases such as New York or Los Angeles challenges the computational limits of existing hardware—and searches must be done in real time while callers are holding on the line.

Our directory assistance automation platform is currently deployed by Fonecta, the largest phone company in Finland, and is in various trials stages with every major US phone company. It's been fun and challenging to learn about a new technology and its applications, especially from the other side of the fence, where I get to build new partnerships, drive requirements to R&D, and manage technical integrations with partners. Many of my co-workers think our CEO is difficult to work for since he frequently calls for status updates (and to check if we're in the office); but it's not a problem for me since I was prepared to work for a taskmaster by Andy as a CS15 Head TA! I've also appreciated the support and networking of Brown CS faculty and alumni for my Internet startups. I'll probably return to development at some point, but for now, I enjoy product management, and I get my coding fix writing stock and sports analysis programs and play programs for my son, Eddy (Brown CS '24?). Eddy already likes to play on the computer, and now that he can walk and talk, programming is just around the corner! Email is welcome at *emb@alumni.brown.edu*.

changelog

'overqualified,' you would have been selected. We would like to offer a sort of consolation prize: Your very short abstract will appear in the ICSE Proceedings, and we would value your attendance at the Symposium to participate as an informal mentor. You will have free admission to the full ICSE conference just as those accepted for the Symposium do."

------8<------

*We're pleased to announce that we've received a gift of 15 Tablet PCs and associated software from **CS Industrial Partner Microsoft Research**. These will be set up in a Tablet PC lab for undergraduate education, which will be used by Andy van Dam's ReMarkable Texts project and Tom Doeppner's Electronic Student Notebook project. Staff members Loring Holden,*

CHRISTOPHER ELAM, Public Policy ’99

While at Brown I spent much time in CS classes and as part of the Graphics Group. Since school I have pursued my interests in dance, receiving an MFA in Dance at NY Tisch in 2000. I have continued to develop my own company, the Misnomer Dance Theater, which was founded in 1998, in my senior year at Brown. I often argue that object-oriented programming has actually become a useful metaphor for some of my choreographic techniques. “Computer science and dance are closely knit in my mind” is what I say to people wondering about this coupling. Since its formation in 1998, Misnomer Dance Theater has found tenderness, humor, and absurdity in people’s efforts to relate to one another. As Artistic Director, I challenge traditional representations of romance, compassion, and closeness by creating moments in which people connect in the most unusual ways. Synthesizing my extensive training in both Balinese and modern dance, I hope to establish a cultural exchange that happens in the body itself. The tension and sculptural definition of Balinese dance is integrated into the fabric of my broken-flow movement. I chose “Misnomer,” the misuse of one word for another, to describe the way a relationship or assumed role may evolve into something unanticipated. *misnomerdance@hotmail.com*

Above: Chris and Andros Zins-Browne ’03; below: Jocelyn Tobias and Chris

CINDY GRIMM, PhD ’96

Bill Smart (PhD ’02) and I got married last summer—twice, once in Scotland and once in Oregon. We both joined the faculty at Washington Univ. in St. Louis two years ago. We’ve found Wash. U. to be very like Brown University, especially the Computer Science department. The students are bright, dedicated, and a pleasure to work with.

We had a very busy summer, since in addition to getting married we had our first exhibition of “Lewis, the Robotic Photographer” at SIGGRAPH. Lewis is a red, four-foot tall robot that takes pictures of people at parties. The project has been (rather unexpectedly) very popular with the press, including the BBC and CNN. The students think this is great because they’re working on a project that their mothers can read about in the local paper!

Last summer I also wrote an NSF CAREER proposal, which, I am pleased to find out, has been funded. The proposal’s title is “A Composition System for Computer Graphics”. The idea is to combine my background in art with my interests in computer graphics to create tools that will help the “average user” make more comprehensible images. Artists learn a large number of techniques for making images easier to understand; can we find ways to make these techniques available to non-artists?

JOHN MARTIN, ScM ’83

Hi Andy: It has been a long time, and you’ve had thousands of students, so you may not remember me. I received my ScM in 1983 and then went to work for your grad school officemate, Dick Wexelblat, at two companies.

I read your address in the latest issue of ***conduit!*** last night, and it struck me that it is time to thank you for your contributions to my education. Even though I was only at Brown for a year, the time was very concentrated and intense. I was used to working hard before I got to Brown, yet you still managed to challenge us in a new way. I remember sleeping on the concrete

changelog

Tim Miller, and Bob Zeleznik and undergrad Yotam Gingold are participating in the former; grad students Roberto Almanza and Qiang Ye and ugrads Albert Huang, Ulises Ortiz, Luke Peng, and Chris Wells are participating in the latter.The tablets will make their debut in Doeppner’s networking course (CS168) at the end of this spring semester and will appear again in his operating systems course (CS167) and van Dam’s graphics course (CS123) this fall. We very much appreciate this support from Microsoft Research.

------8<------

*The 03-04 academic year is the department's 25th! We're planning a special distinguished lecture series to honor this event as well as a number of other festivities, including a celebration on **May 27, 2004**, the Thursday before*

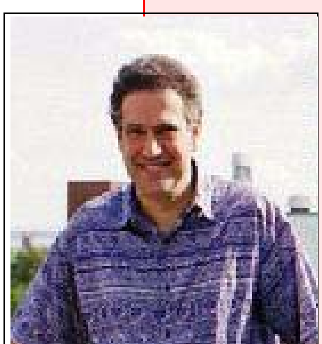
floor of the Cause Lab, face down, in between tracking down bugs. What we built as a project in your course was comparable to what Yahoo Maps does right now, and I am so proud that my team built that in a semester project.

On a personal level, I am still very good friends with people I met in your graphics class 20 years ago.

I have been working at NYNEX/Bell Atlantic/Verizon for the past 12 years. If I look out of my building, I can see the building where Blue Sky Studios created the movie "Ice Age". I imagine that there are some of your students there right now creating the next big animated film.

It seems like only yesterday that I visited Brown CS for the first time, and you spent a lot of time with a few of us prospective students in your office. I was sold at that moment, and going to Brown was one of the smartest things I've done in my life. Thank you. John Martin. *martin@basit.com*.

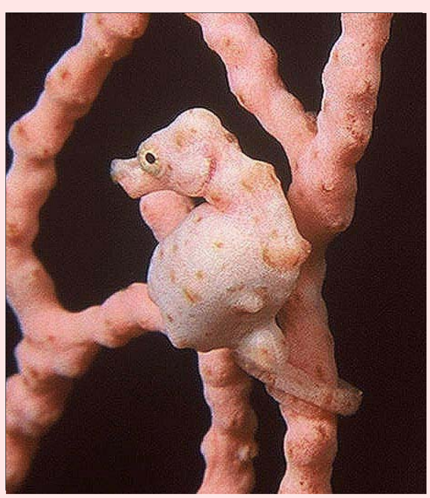
PYGMY SEAHORSE



The photo is of a pygmy seahorse (Hippocampus bargibanti), taken last August off the island of New Ireland, Papua New Guinea at a depth of 60 feet. Such seahorses are quite small (this one is less than a quarter-inch long) and incredibly hard to find—they look exactly like the coral they live on. They were discovered only after someone collected some coral, put it in an aquarium, and then (finally!) noticed the tiny seahorses, probably dead by this time. Knowing that the coral at this particular location had some pygmy seahorses on it, it took my two dive guides and me 10 minutes to find them. They were once thought to be rare, but, now that divers know to look, they are turning up at numerous sites in the South Pacific.

For the camera literate, this was taken using Fuji Velvia film and a Nikon F4 in a Nexus housing at an aperture of f22 at 1/250 second with two large Ikelite strobes through a 105mm macro lens at closest focus with a closeup adapter increasing the magnification by 90%.

Tom's newly updated web page with many recent underwater photos is at: *http://www.cs.brown.edu/people/twd/fish/start.htm.*



changelog

Commencement. Please keep this date open!

------8<------

As you can see from Don Stanford's CS002 spring weekend Hawaiian shirt

extravaganza, they seem to have raised the bar on this event!

------8<------

Renovations to the CIT 3rd floor will begin in June so CS can expand into it by September. This means we can admit more graduate students and have space for more faculty and visitors.

------8<------

CS is on the main Brown web page again; that's grad student Dan Keefe in the Cave! www.brown.edu

NEWS FROM KEITH ADAMS '00

Much has changed since my graduation only three years ago. When I packed up my Providence apartment to join a small virtualization software company out west, I left behind a Brown very different than the one present students inhabit, and I was entering a very different world than the one greeting the class of 2003.

2000 was the Internet Boom's swansong year. Every CS graduate who cared to could paper a bedroom wall with job offers. Brown's Sun lab actually contained Sun computers. A departmental Berlin Wall split the systems lab in two, and the inhabitants of Syslabs A and B regarded each other with suspicion and distrust. Many CS courses were crowded with students who seemed to have a passing interest in computation but a profound interest in making six figures while hacking javascript. And those of you expecting to graduate in 2003 were freshmen, meaning that the modicum of face recognition I still enjoy on recruiting trips back to the CS department will soon be gone.

...Brown consistently produces some of the best computer science graduates in the world. I knew abstractly in 2000 that I'd gotten a great education at Brown, but working at VMware has truly driven the point home

VMware will keep sending me back, though, because Brown consistently produces some of the best computer science graduates in the world. I knew abstractly in 2000 that I'd gotten a great education at Brown, but working at VMware has truly driven the point home.

When Matt Eccleston ('00) and I showed up at VMware's anonymous office space in the summer of 2000, we were among the first engineers employed there without graduate degrees. For most new college grads, being both so new and so junior would have been intimidating. This pair of veteran CS169 TAs, though, was uncowed by the mix of professors, PhDs and PhD program dropouts. After all, we would have TAed some of these people back at Brown.

VMware, the company, is increasingly hard to describe concisely. We're a system software company that began as an outgrowth of Mendel Rosenblum's virtual-machine research at Stanford. For those who were napping that day in 167, in the '60s and '70s virtual machines enabled mainframe users to make huge IBM boxes look like multiple, smaller machines. Even though computers are cheaper today than in the mainframe's heyday, virtual machines remain useful, perhaps more so than ever. Lest I lapse into starry-eyed marketspeak, I'll refer those curious about the applications of VMware';s products to our web page (*www.vmware.com*).

My three years here have been deeply satisfying. VMware has been successful during very difficult times, growing from a rock'em, sock'em startup with 80 employees to a more stable, though small, company of 300. We've gone from shipping a single product to shipping three major products (Workstation, GSX Server, and ESX Server). Matt and I both played important roles in the bringing the world's first multiprocessor virtual-machine product to market. Across the board, we have been entrusted with much more responsibility than larger companies would ever consider giving a pair of young punks straight out of school.

We've also been joined by fellow Brownistas Paula Robin ('01) and Rob Manchester ('??), with hope for a still unfinalized handful of folks to follow this summer. And we hope to see more in the future, for Brown CS and VMware, like chocolate and peanut butter, are two great tastes that taste great together.

Of course, a Brown degree is no guarantee that a person is bright, talented, or motivated. My year certainly contained its fair share of unredeemable goofballs; I'm sure you think the same is true of your year. However, having been involved in recruiting from other ostensibly top-tier CS schools, I think that Brown is weirdly good at nurturing those with the potential into successful practicing computer scientists.

I'm not sure why this is. So much is atypical about the whole Brown CS experience that it is difficult to match causes to effects. Is it the undergrad involvement in TAing and research? The faculty's admirable focus on undergrad education, rare at a research university? These certainly play a part. But I'll go out on a limb and claim that, for at least some students, it's mostly the coding.

Now, there are those who suspect that any programming in an academic class is *sub rosa* job training. I understand this fear; I recall with a shudder those javascript jockeys who flooded my classes a few short years. I am not advocating that we turn Brown into a trade school. Rather, I think that for people of a certain psychological makeup, coding is important purely as an *educational* tool. When I was a student, the little blip of seratonin that came from translating lecture slides into a functional lump of software helped cement those lecture slides in my brain. There are precious few other top-rank CS schools that offer the opportunity to learn by doing as deeply and broadly as Brown, and my education would have been poorer, or at least less enjoyable, anywhere else.

It's possible to go overboard, of course. Some material is hard to teach in an applied style, and 152 rocked, though I didn't write a line of code for it.

Thus far, I must rate my transition from Brown CS student to productive member of society a success. Many, many Brown CS folks end up in the San Francisco Bay Area, so the social blow of relocation has been greatly cushioned. I considered listing all the Brown CS people I'm in contact with out here, but it runs to 40 or so. It seems at times to be a veritable CS department-in-exile. When I want to wax nostalgic about elevator keys, prank-calling the Sunlab consultant, getting wings and watching "Shaft", or watching those three blinking red lights from the pink couches on the fifth floor at night, there's always a fellow Browniac nearby, ready to hold my hand down memory lane. *kma@vmware.com*.

RARE TROPICAL VISITOR

From January 9 until March 14, seven beautiful Zebina Hairstreak butterflies mysteriously appeared one at a time in Suzi Howe's office! There are no flowering plants in the office, so Suzi was feeding them sugar water and deposited a couple next door in Trina Avery's office, which happened to have some plants in bloom. Each survived from three to five days. From what we have gleaned from on-line sources, our butterflies were all females, their grey wings outlined in black with two additional orange, black and white lines defining the fore and hind wings and creating a streak (hence the name). Orange eyespots with black dots, the better to confuse predators, were located on the hind wing, where the wing shape changed to an elegant bracket curve with delicate white-tipped appendages trailing behind. These eyespots were on both sides of the wings and dramatically visible when the wings opened flat. When the sun shone directly on these little guys, the edges of their wings glowed a neon blue...we were definitely smitten!

*According to **Butterflies of North America**, this little-fingernail-sized visitor is considered a 'rare tropical stray.' It ranges from Colombia to Mexico and only three such strays have been recorded in southern Texas, which makes our Rhode Island visitation all the more thrilling. Well, maybe not for everyone—when astaffer Fran Palazzo saw one she said, "That's no butterfly, that's a moth and it's going to eat my sweater...where's the fly swatter?"*

*After Suzi emailed an emergency posting asking for a lepidopterist in the department to help solve the mystery, Shriram Krishnamurthi noted that he and his wife had just returned from a trip to Mexico over winter break and the hairstreaks might have flown out of their luggage...he admits this is a 'wild speculation' but suggested we should give a small award to the person who posits the most extravagant theory—a free annual subscription to **conduit!** (runner-up gets two!)...*

LIDDY SHRIVER '90

Elizabeth (Liddy) Shriver's father Bruce, a pioneer in computer architectures and microprogramming in particular, met Andy van Dam in the '60s. Since they have been friends so long, Andy was delighted when Liddy came to Brown. She worked under Jeff Vitter and received her M.Sc. in 1990; her thesis topic was "Optimal Disk I/O with Parallel Block Transfer." After earning her doctorate at NYU, she went to work at Bell Labs.

Practicing for the Denmark tour

Liddy's health was normal until nine years ago when she became aware of a burning pain in her foot. After years of this severe pain, a tumor was found behind her knee, which a biopsy showed to be Ewing's sarcoma/PNET (primitive neuroectodermal tumor). Says Liddy, "I had chemotherapy, surgery, radiation, and more chemotherapy, but my cancer has spread to my lungs. So far, the nodules in my lungs haven't responded to treatment." As the disease progresses, Liddy will have increased difficulty breathing, which will hinder her great passion, cycling. Before this happens, she plans to take several long bike rides as part of a campaign to raise public awareness of sarcoma, especially to the critical lack of young adult participation in clinical trials. Because most Ewing's sarcoma research has been done on children, the results do not always apply to adults. This lack has resulted in a higher fatality rate for adults. "I fall into the category of *young adults with cancer*," says Liddy, "which means there isn't much data available to help my oncologist know what treatments might work for me."

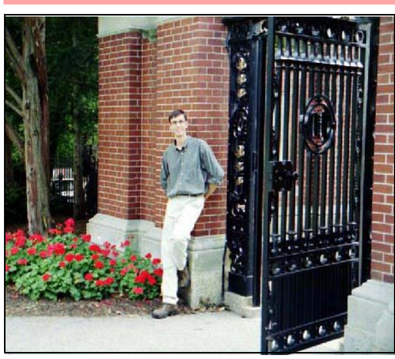
On April 10-13, Liddy, 36, joined friends and family members for their first **Team Sarcoma** bike ride at Cycle Zydeco, the second annual four-day, 200-mile southern Louisiana bike tour. In late June and early July, Team Sarcoma will undertake a bike tour in Denmark. To join the team or for more information, please see the following websites or contact Liddy at *liddy@theshrivers.us*.

www.theshrivers.us/la_bike_our.htm

www.theshrivers.us/danish_bike_tour.htm

www.cyclezydeco.com

MOTION CAPTURE



Michael Black

The goal of the human motion project at Brown is to develop algorithms that can "capture", or estimate, the motion of people in video sequences. This problem is challenging due to the variability of human appearance, the complexity of the human body and its motion, and the loss of three-dimensional information when a 3D scene is projected onto 2D images. If we can solve the problem there are applications as varied as animation, medical diagnosis, and human-computer interaction. To study this problem we are building a one-of-a-kind motion-capture facility at Brown.

The first component of this facility is a commercial Vicon motion-capture system. This system uses six cameras (right) and infrared illumination to detect and track reflective markers placed on the body of a subject (this is the same system used by Hollywood to transfer the motion of human actors to animated characters).

Using the six cameras and special-purpose hardware, we collect the motion of various subjects performing a variety of tasks. Machine-learning techniques are then used to model how humans move.

Vicon cameras with ring of infrared LEDs

Temporary motion-capture lab

Capture session. Note reflective markers on the subject

This system, however, is not fully general since it requires that people wear special markers. We would like a markerless system that "understands" something about how people appear in images. To that end we have installed a second video-based capture system that can save the video data from four high-speed progressive-scan cameras. We are in the process of synchronizing these two systems so that we have video sequences of people from multiple views along with the "ground truth" 3D motion provided by the Vicon system. This will provide a state-of-the-art facility that can be used to evaluate and compare current human motion estimation algorithms and to provide new types of training data for machine-learning approaches to human motion analysis.

This work is supported by the DARPA HumanID project. The Vicon system was also partically funded by the NSF IGERT program and is shared with the VENLab in the Department of Cognitive and Linguistic Sciences.

fac.activities@cs.brown.edu

MICHAEL BLACK. Michael gave a number of invited talks on brain-machine interfaces, including the keynote talk at EURON, the European Robotics Research Network annual meeting in Lisbon; a plenary talk at the Ohio State Mathematical Biosciences Institute workshop on neural coding; and a plenary talk at the First International IEEE EMBS Conference on Neural Engineering in Capri. After many years of waiting he had a patent finally issued on work done at Xerox PARC with Allan Jepson from the University of Toronto ("Apparatus and method for identifying and tracking objects with view-based representations"). Siemens Corporate Research gave a generous $25,000 gift to support his research on human motion analysis in video sequences.

ROGER BLUMBERG. Roger has been named to the Core Technology Group of the International Virtual Medical School Project (IVIMEDS.org), an international collaboration of more than fifty leading medical schools and universities worldwide including Brown's School of Medicine. The Technology Group is involved in the development and evaluation of the reusable learning objects (RLOs), instructional software, and other e-learning aspects of the IVIMEDS project.

UGUR CETINTEMEL. Ugur served on the program committee of the International Conference on Distributed Computing Systems, held in Providence this May. He also worked on the organization of the Workshop on Internet Applications to be held in San Jose later this year. Ugur gave talks at the Symposium on Reliable Distributed Systems in Osaka and the Conference on Innovative Data Systems Research. This semester he started teaching his new course, Networked Information Systems (CS 138), a hands-on introduction to fundamental principles and practice underlying today's networked information systems.

AMY GREENWALD. Amy has received one of this year's Richard B. Salomon Faculty Research Awards, established to support research projects of exceptional merit. Amy's project will compare human decision-making processes about correlated equilibria with the reasoning of computational agents.

SHRIRAM KRISHNAMURTHI. Shriram spent part of winter break in Mexico. His highlight was visiting Teotihuac'an, which he had read about for nearly 20 years but never seen before. He found it mildly ironic that he had to move from Texas to Rhode Island to get a chance to cross the border.

In the fall, Shriram picked up two more award paper nominations, one each from Automated Software Engineering and Foundations of Software Engineering. He was especially thrilled that both papers had a Brown undergrad, Harry Li, as a co-author. Harry's now in the PhD program at UT Austin. Shriram also served on the program committees for Programming Language Technologies for XML, Lightweight Languages 2 and Practical Aspects of Declarative Languages.

January was Shriram's month for invited talks. His Mexico visit was instigated by an invitation, to him and three book co-authors, to run a programming languages summit at the Universidad Nacional Autónoma de México. The next week he gave a keynote talk at Practical Aspects of Declarative Languages, and the week after that he spoke in the Distinguished Seminar Series at IBM Watson. He's very glad nobody's inviting him to anything else right now.

Dan Keefe (drawing) and Cullen Jackson (behind camera) working on a user study investigating how graphic designers and illustrators critique and create scientific visualizations. Here Dan Keefe is creating a new data-driven scientific visualization based on his analysis of other visualization techniques previously viewed

DAVID LAIDLAW. In April David was awarded the prestigious Henry Merritt Wriston Fellowship for '03-'04 in recognition of his innovative Brown/RISD course with Fritz Drury of RISD's Illustration Department. CS and RISD students work together in teams to develop new ideas for the visual presentation of fluid-flow data, using traditional 'crits' to evaluate and refine their designs. David is now an associate editor of IEEE **Transactions on Visualization and Computer Graphics**.

FRANCO PREPARATA. Franco recently spent a few weeks at the National University of Singapore as the Qwan Im Thong visiting professor of Computer Science. He was recently appointed to the Scientific Board of the Istituto per la Scienza e la Tecnologia of the National Research Council of Italy and to the International Advisory Panel of the Graduate School Integrative Sciences of the National University of Singapore.

JOHN SAVAGE. John finished his third year as an officer of the Brown faculty and a member of the Faculty Executive Committee and a year as a member of the Academic Priorities Committee, a new committee introduced as part of a major overhaul of faculty governance spearheaded by the Task Force on Faculty Governance, which John chaired. The Task Force, which began its work about a year ago and finished in early April, introduced changes to increase collaboration between the faculty and the administration and reduce the number of committees from 44 to about 25 and the number of faculty slots from 237 to about 135. Last summer John finished eleven years of service on the visiting committee for the Department of Electrical Engineering and Computer Science at MIT, his alma mater.

DON STANFORD. Don has been busy teaching CS002, which has a record 317 students enrolled! In addition, he introduced GTECH to a *legal* form of gaming for the Internet, which will let them create a product that appeals to Internet-centric players.

ELI UPFAL. Eli was an invited speaker in a workshop on Topics in Computer Communication and Networks at the Isaac Newton Institute for Mathematical Sciences, Cambridge, UK, and served on the Program Committee of WEA 2003, the Second International Workshop on Experimental and Efficient Algorithms, held in May in Ascona, Switzerland.

PASCAL VAN HENTENRYCK

Pascal was awarded the 2002 INFORMS ICS prize for research excellence at the interface between operations re-search and computer science. This prize, given annually since 1986, recognizes Pascal's many contributions to constraint programming and its integration into operations research. Pascal also gave invited talks or tutorials at the annual INFORMS meeting in San José, at the IN-MA school on large-scale optimization in Minnesota, at the Optimization Days in Canada, and at the New England Symposium on Programming Languages and Systems. He was also program chair of the International Conference on Constraint Programming, which attracted a record number of submissions. His brother and father-in-law made a surprise visit on March 8 for his 40th birthday, coming all the way from Belgium.

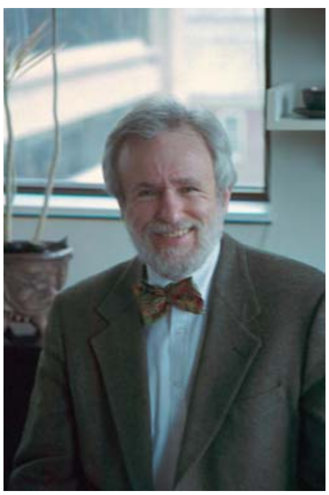
EILEEN VOTE. Eileen, a postdoc working with David Laidlaw, has been awarded one of this year's Richard B. Salomon Faculty Research Awards for research projects deemed of exceptional merit. She will work on a three-dimensional reconstruction of the water system at Qumran, the find site of the Dead Sea Scrolls. A digital reconstruction of the water system will help derive a chronology for the archaeolo-

l to r: Cullen Jackson, RISD Professor Fritz Drury, David Laidlaw, Daniel Acevedo and Dan Keefe meet in the atrium to discuss the critical features of data-driven scientific visualizations and how to combine them into perceptually optimal images gy of ancient Qumran, thus providing a context for insight and discovery about the Dead Sea Scrolls.

PETER WEGNER. Peter's article in the April 2003 **CACM** examines the impact of Turing machines on mathematics and the theory of computation. His article in the forthcoming book on the 90th anniversary of Turing's birth in 1912 examines the impact of Turing's work on current computer science.

A film by the Brown Medical School about Peter's remarkable recovery from his serious accident four years ago is being used by medical students to examine medical and ethical issues in helping people recover from brain injuries. His work as editor of the Faculty Bulletin allows faculty to write about improving teaching and research at Brown. He uses his office for reading and research most weekdays, appreciates the department's thoughtfulness in supporting his retirement and is working with Lew Lipsitt on Brown's upcoming retirement community.

CHARNIAK UNPLUGGED



Eugene Charniak

A few months ago CNN did a piece on the 25th anniversary of the smiley ":-)", now the ubiquitous indicator of email emotional state. I was surprised to see a colleague of mine, Scott Fahlman at CMU, explaining that, as far as he or any of his colleagues knew, he was the first person to use a smiley, and that he did so in an email sent 25 years ago. I have known Scott for somewhat longer than that, as we overlapped as MIT grad students in the early '70s, and since his field is also artificial intelligence we have frequently met at conferences ever since. However, I had had no idea that he was so distinguished!

The CNN interview was pretty funny. They asked him if he ever got any financial reward for this major invention. He responded that he had not, and to the follow-up question of whether he wished he had, he responded with good humor that if he got money for every use of the smiley, people would probably use them less, and he would rather see them in constant use. At any rate, I sent him some email congratulating him on the 25th birthday of his invention; in his response he said that the CMU publicity people had had the idea of publicizing this bit of web trivia, that it had taken off much more than anyone expected, and that he was glad things were dying back down so he could get some work done.

A few days later I mentioned all this at a lab meeting, and one of my graduate students, Heidi Fox, said that SHE knew the fellow who invented the use of the "@" in email! I quickly realized that this was surely not an idle boast. Before she became a full-time graduate student Heidi worked at BBN, the company that had had a major role in the construction of the so-callet "ARPANet," the predecessor of the Internet and ARPA's (DOD's Advanced Research Project Agency) major claim to popular relevance. At any rate, it was clear that I had been one-upped, since as even Scott Fahlman would have to admit, the "@" is more critical in current daily life than the ":-)".

Many ***conduit!***s ago I explained how it came to be that PhDs in computer science here get rubber chickens thrown at them at the reception in their honor immediately after their thesis defense. Some time later I mentioned that Mary Harper, one of my students, is now giving rubber chickens to her PhD students. A few weeks ago I learned that another of my students, Sharon Caraballo, has also been engaged in some fowl play (I'm sorry, I am not strong enough to resist). Sharon took her chicken to a taxidermist and had it stuffed and mounted! It is now proudly displayed on a wooden plaque above her desk at her office at Georgetown. She explained that the wooden plaque is actually one for fish, since the rubber chicken is too small for anything else. She also told me that when she took it to the taxidermist she gave it to him with a straight face, no explanation. I can only try to imagine what the guy was thinking. However, she was accompanied by her husband when she went to pick it up, and her husband took pity on the fellow and told him what was going on. The taxidermist said he had probably gotten more comments on it than anything else he had ever done.

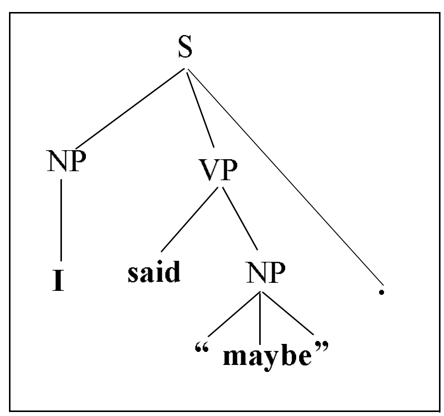
Having just defended his thesis and received his PhD poultry, Air Force Major Joel Young is a happy man!

In case you want to have your chicken mounted, you might want to take it to someone with experience, so Sharon has given me the fellow's address:
Taxidermy Unlimited
Vienna, VA
703-255-1460
Tell him "***conduit!*** sent me."

Those of you not interested in details of the English language can skip the rest of this article, as I feel the need to fess up to some ungrammaticalities in the first three paragraphs of this edition of "unplugged." As is well known to all copy editors but only moderately known to the rest of us, commas and final punctuation go inside final quotation marks, not after, as placed in the previous sentence. This can be a real bother. So in the first paragraph I had a situation where my smiley ":-)" came just before where a comma should occur. Thus I should have put the comma inside the quotes, but I did not, because ":-)," would not be the correct smiley. That is, I should not have written "":-)",," but rather "":-),"." (Everyone got that?)

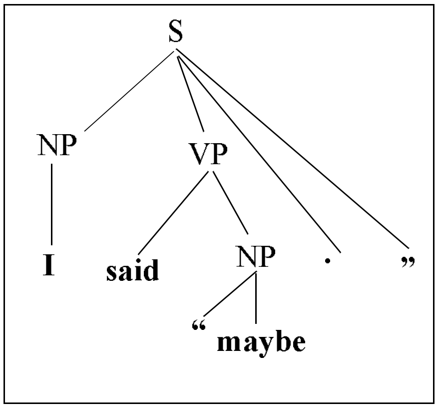
This convention, even when applied to normal text, hardly ever makes sense. Consider the sentence: I said "maybe." As I have noted in previous ***conduit!***s, I have been doing research on getting computers to establish the correct sentence structure for sentences such as that. This involves creating tree-structures that explicitly indicate structure in terms of noun-phrases (NPs), verb phrases (VPs), etc. Now if English were logical, we could have written the above sentence as: I said "maybe".

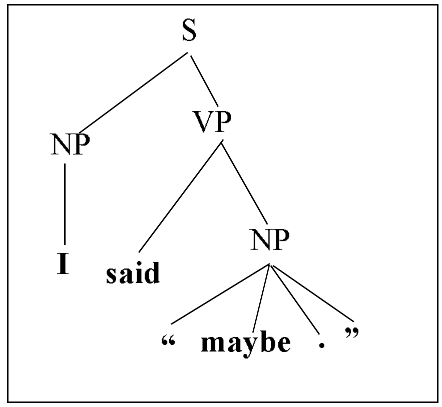
Then we could nicely write out the tree-structure as follows:



Note how the quotation marks go with the noun phrase (they help describe what was said) while the final punctuation goes with the complete sentence (S), as it should.

However, with the nonlogical version we are left with two very unpalatable alternatives:





In one case the NP ends up having only a single quotation mark and the S gets the other, while in the second case the NP gets both quotation marks but also the period, which makes no sense.

Now this may strike you as completely unimportant, but I plead with you to see my point of view! To get my parser to work I use a large corpus (one million words) of hand-annotated syntactic structures. This corpus, for lack of anything better, uses the first of the above terrible messes. However, I am now trying to apply this work in machine translation, and since other languages do not share English's mess, we encounter a problem when trying to figure out how quotations should be translated. This causes no end of grief. The English convention has to be reformed. And, you can quote me, “immediately!”

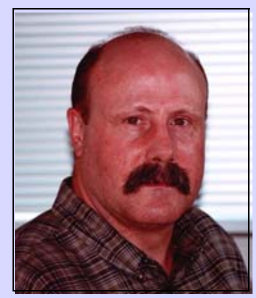
conduit!

A publication of
The Computer Science Department
Brown Uiversity

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

Suzi Howe
Editor-in-Chief

Katrina Avery
Editor



Jeff Coady
Technical Support

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

conduit!

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

Printed on recyled paper

Address changes welcomed