Conduit

Autonomous Bidding Agents:

Strategies and Lessons from TAC Travel

Also inside:

→ Evolution by Intelligent Design: the New CS Concentration Requirements

→ Nine years of CS17-18

Industrial Partners Program

The primary goals of the Industrial Partners Program (IPP) are to exceed the expectations of our partner companies in terms of recruiting and outreach; to allow our faculty to engage in challenging and meaningful research collaborations; and to provide resources, funding, and employment to our students.

The department wishes to thank our Industrial Partners for their support:

Collaborators
Cisco Systems

Premier Partners
Adobe
Network Appliance
Sun Microsystems

Affiliates
Apple
Google
GTECH
ITA Software
Oracle
Pfizer
VMware

Small Business Supporters
IAM Technologies
Linkage Systems
StreamBase Systems

Individuals
Jim Baker, Bivio Software
Paul Edelman, Edelman & Associates

For more information about the Industrial Partners Program, contact:

John Hughes, Faculty Director
Telephone: (401) 863-7600
John_Hughes@brown.edu

Amy Tarbox, Program Manager
Telephone: (401) 863-7610
ipp@cs.brown.edu

To learn more about IPP visit:
www.cs.brown.edu/industry/ipp/

Notes from the Chair: the Latest News from 115 Waterman

Greetings to all CS alums, supporters and friends!

The 2007 academic year is coming to a close and the department is full of activities. As I am writing these notes we are in the midst of a very ambitious faculty hiring season. We are interviewing promising candidates in the areas of computer systems, machine learning, and computer graphics. The department is also involved in the faculty search to fill a new faculty position in the Center of Computational Molecular Biology. We are hoping to fill 2-3 positions this year and another position next year. These hires will conclude a ten year expansion period in which the number of faculty in the department has almost doubled, bringing new breadth and depth to the department research and teaching.

It is always a pleasure to report recent honors and awards received by our faculty and students, and the list, as usual, is long and impressive:

Professor Stan Zdonik was named Fellow of the ACM for his contributions to data management and database systems.

Professor Roberto Tamassia won the 2006 IEEE's Computer Society Technical Achievement Award for his pioneering research in the field of graph drawing and for outstanding contributions to the design of graph and geometric algorithms.

Assistant Professor Ben Raphael was included in the *Genome Technology* magazine annual "Tomorrow's PIs" list. The thirty selected researchers, who are peer nominated, must be involved in the disciplines that comprise systems biology and within five years of their first faculty or equivalent post.

Meinolf Sellmann is the most recent junior faculty member to receive a National Science Foundation (NSF) CAREER grant. His is for the proposal of the Cornflower Project which he describes in this issue of *Conduit*.

Assistant Professor Ugur Cetintemel has been promoted to the rank of Associate Professor with tenure, effective July 2007.

In October the department hosted a very special event: "Excursions in Algorithmics: A Late Festschrift for Franco P. Preparata" aka FrancoFest, a two-day workshop to honor and celebrate the career of Franco P. Preparata on the occasion of his 70th birthday (which was in December 2005). You can read more about these lavish events on page 18.

I have also two student awards to report: Ph.D. candidate Danfeng Yao received the Best Student Paper Award at the Eighth International Conference on Information and Communications Security (ICICS '06) for her paper titled, "Point-Based Trust: Define How Much Privacy Is Worth." The work was in collaboration with her advisor, Roberto Tamassia, Keith Frikken at Miami University, and Mikhail Atallah at Purdue University.

Undergraduate students Daniel Leventhal and Leo Meyerovich have received Honorable Mention awards from the Computing Research Association (CRA). CRA's Outstanding Undergraduate Awards program recognizes undergraduate students who show outstanding research potential in an area of computing research.

Alumni Outreach: Be sure to mark Saturday, May 26, 2007, as the date of the next Computer Science Reunion and Networking Reception. We encourage all alums, friends and supporters to stop by. As those who attended the previous year's event can attest, it's certain to be a fantastic time! Registrations for the reunion can be made online at: http://www.cs. brown.edu/events/reunion/home.html

Finally, I'm very happy to report that I'll be stepping down as a department chair at the end of this academic year. Next I plan to spend a productive (and tasty) sabbatical year with my family at the University of Padova, Italy.

Best wishes, and very good luck to the next department chair to be named shortly.

Keep in touch!

Eli

Spring|Summer 2007

FEATURE

Autonomous Bidding Agents: Strategies and Lessons from TAC Travel

Amy Greenwald...PAGE 8

Evolution by Intelligent Design: the New CS Concentration Requirements

Tom Doeppner is interested in operating systems and everything related to them. He is currently interested in the area of operating system support for security and is investigating means for running arbitrary programs without fear of the consequences.

Once again, we've changed the CS concentration requirements—this time for the A.B. and Sc.B. degrees, with adjustments to the joint concentrations to follow. The changes, not earthshaking, are an attempt to keep our requirements in line with the consensus educational philosophy of the CS faculty, within the framework of the Brown open curriculum.

The curriculum committee (chaired by John Savage with members Chad Jenkins, Claire Kenyon, Franco Preparata, Steve Reiss, and me) met weekly most of last year. Everything was on the table. In what follows, I outline current requirements and discuss the new ones.

The current curriculum has a six-course core requirement common to both the A.B. and the Sc.B. It consists of an introductory sequence, either 15/16 or 17/18. Though they have different approaches, both sequences get students to the same place. After the introductory sequence, students take a two-semester introduction to what might loosely be called computer systems. First they take CS 31, covering the basics of digital logic, computer architecture, and systems software such as compilers and operating systems. For the next course, students must make a choice between CS32 and CS36. The former is an introduction to software engineering featuring modern high-level techniques, including more advanced object-oriented design, multithreaded programming and XML, as well as additional modern software engineering practices. The latter is an introduction to systems programming, featuring techniques and concepts of systems programming, as well as how to program in C and C++ (see my article in the spring 2005 *Conduit*). All students take a pair of courses preparing them to study CS theory: CS22, covering discrete mathematics and probability, and CS51, covering the basic theory of computation. In addition, all students must complete two semesters (or the equivalent) of college-level calculus as well as an English writing course.

Tom Doeppner

Students pursuing an A.B. must choose two advanced courses from one of the following areas: software systems, theoretical computer science, artificial intelligence, and systems (including hardware). If you've been counting, that's a total of eleven courses (though the calculus and writing courses are considered prerequisites for the concentration rather than part of the concentration itself).

Students pursuing an Sc.B. must complete two Math or Applied Math courses beyond calculus, as well as a two-course sequence in some science other than CS. The math is required because it's extremely useful, sometimes essential for pursuing many advanced topics in CS. The science is required because we want our Sc.B. concentrators to be able not only to apply their CS to another area, but also to bring the approaches of other areas into CS.

Sc.B. students must complete seven advanced courses in CS. They must choose one course from each of the areas software systems, theoretical computer science, artificial intelligence, and hardware systems. One of the seven courses must be a design or independent-study course (this is a university requirement that's rather ill-defined in our curriculum). Though neither CS32 nor CS36 is designated advanced by its number (advanced courses are numbered 100 and above), we allow one of them to be used as such if the other was used for a core requirement, since we believe that anyone going into a career involving software should learn the material of both. Again, if you've been counting, the total number of courses required for an Sc.B. is twenty (out of thirty required by Brown to graduate).

That's the current concentration; what's new?

We had much debate about the common core of the Sc.B. and the A.B. Some of us felt that we should be more flexible and allow students, particularly those pursuing an A.B., to pick and choose the courses that made sense for their goals. Others of us felt that, at this early stage in students' education, they don't really know enough yet to make meaningful choices. Furthermore, most of us feel there are many topics that should be understood by anyone graduating with a CS degree from Brown. We decided to leave the core pretty much as is. Students already may choose between CS32 and CS36. We will be allowing alternative CS theory courses other than CS51 as soon as we add them to our curriculum.

We still believe that the calculus and writing requirements are important. Two semesters of calculus are required in pretty much all sciences, partly because it's used in more advanced courses and partly because without calculus, one really cannot go further in mathematics. Though many computer scientists never encounter a derivative or an integral after their freshman year, many of us believe that no one can call themselves well educated without at least this much calculus. Furthermore, as computer scientists become involved in cross-disciplinary work with other sciences, math is becoming increasingly important.

The English writing requirement has been on our books since the early 1980s and its justification hasn't changed. What's involved in organizing one's thoughts and putting them on paper differs almost not at all from designing software or structuring a proof. All must be done well and all are essential for success in computer science.

For the A.B. degree, we feel that students should "get depth" in at least one area, where depth means at least two courses, and also that the broad areas of the current requirements are a bit too amorphous. So, we've put together a set of pairs of related courses: students must take such an approved pair to satisfy our depth requirement. We also feel that the current A.B. is too weak—two

"As computer scientists become involved in cross-disciplinary work with other sciences, math is becoming increasingly important."

advanced courses are not much—so we've added a third advanced course that may be in CS or any related area. Students may use this course to go into further depth in their chosen area or to broaden their education. Indeed, for some areas we feel that a good education requires a particular third course. Thus, for example, students taking a course pair involving computer systems or

software engineering must take either CS32 or CS36 as the third advanced course, whichever was not used in the core. The number of courses required for the A.B. has gone up from eleven to twelve—a modest increase and within university guidelines for an A.B. degree.

We've stayed with twenty courses for the new Sc. B. requirements but have changed the distribution of advanced courses. We're sticking with requiring seven of them. Three must be in separate areas: CS theory, AI, and CS systems. As with the new A.B. requirements, we want students to get depth in an area by taking courses from approved course pairs; for the Sc. B., we require four of the advanced courses to form two approved course pairs. Last, we've strengthened the requirement for a "design or independent-study course." We want Sc.B. concentrators to have a "capstone experience"— a senior-year project that builds on much of what they've learned at Brown. So we require one of the advanced courses to be such an experience. It must be taken during the senior year and lead to some sort of result or artifact (such as a software system or thesis writeup). It may be independent work or group work.

What's newly absent from the Sc.B. requirements is an Engineering course. Ever since the department began, our Sc.B. concentrators have had to take a course in Engineering, such as digital logic or computer architecture. Though we still feel that such a course is useful, we don't think that it's so useful as to preclude taking other advanced CS courses. So, we've made it optional—students who want to take such a course may do so for concentration credit, but they don't have to.

Our requirements will no doubt continue to evolve. We've made this somewhat easier to accomplish than in the past by the notion of approved course pairs. As courses evolve and are added and dropped, we'll update the list of pairs to ensure that students really do get depth in the areas of their choice. As usual when we change our requirements, current concentrators are grandfathered—they may use either the old or the new requirements. **C!**

Nine years of CS17–CS18

If you haven't been a student at Brown in a decade, you might not know that about half of graduating CS concentrators these days never take CS15 (the course formerly known as CS11) or CS16 (the course formerly known as CS21). As an experiment, the department introduced an alternative introductory Computer Science sequence in 1998, and the "experiment" will be conducted next year for the tenth time.

CS17 and CS18 are officially titled "Computer Science: An Integrated Introduction." The term "integrated" reflects one of the unusual characteristics of the course sequence. According to the traditional approach to computer science education, a student first takes a semester-long course in programming, and then follows it up with a semester-long course in data structures, algorithms, and analysis. The vast majority of educational institutions follow this model. However, this approach has some drawbacks. Here is an excerpt from the Association for Computing Machinery's Computing Curricula 2001, listing some commonly cited drawbacks:

"Focusing on programming to the exclusion of other topics gives students a limited sense of the discipline, thereby reinforcing the common misperception that "computer science equals programming." Theoretical topics that would enhance the students' understanding of the practical material are deferred to later points in the curriculum, when they no longer have the same immediate relevance. This limitation has implications for both majors and nonmajors. Nonmajors who take only introductory courses are deprived of any exposure to the conceptual and intellectual foundations that underlie the revolutionary technological developments driving change throughout society. For majors, the fact that theory is not introduced in the early courses fuels the bias of many students who conclude that theory is irrelevant to their educational and professional needs.

"Programming courses often focus on syntax and the particular characteristics of a programming language, leading students to concentrate on these relatively unimportant details rather than the underlying algorithmic skills. This focus on details means that many students fail to comprehend the essential algorithmic model that transcends particular programming languages. Moreover, concentrating on the mechanistic details of program-ming constructs often leaves students to figure out the essential character of programming through an ad hoc process of trial and error. Such courses thus risk leaving students who are at the very beginning of their academic careers to flounder on their own with respect to the complex activity of programming."

In 1998, I felt that the culture of the undergraduate CS community undervalued the conceptual and analytical aspects of computer science; the dominant viewpoint, especially among students in their first couple of years, seemed to equate computer science with programming. Moreover, I felt we were failing to attract or retain some students who were turned off by the emphasis on the instrumental side of computer science (build something to achieve an extrinsic goal) and who would be drawn to the intrinsic beauty and elegance of many ideas in computer science (an aspect of computer science that for some is just as attractive as the satisfaction of building something to achieve an extrinsic goal). Finally, I believed that our undergraduates' conception of programming—and of programming paradigms—was a bit too narrow.

A student, Sarah Finney, who had helped me teach a course on cryptography for nonmajors (CS007: Secrets and Promises) approached me with the suggestion that I develop an alternative introduction to the major. It occurred to me that such a course might have a salutary effect on the undergraduate CS culture. I enlisted the help of Professor Leslie Kaelbling, who at that time occupied the office next to mine, and we developed and test-drove the course that summer with the assistance of undergraduate TAs Junichi Kimura, Sheryl Koenigs-burg, Adam Leventhal, Phil Levis, Danny Lonborg, Brian O'Neill, and Ph.D. student Don Blaheta.

We co-taught the course that academic year, and the course has been taught every year since. Professor Kaelbling left for MIT, but Professor John Hughes stepped in and has participated in teaching and shaping the course for many years. This semester, Professor Ugur Cetintemel is teaching CS18, and next year, Professor Amy Greenwald is slated to teach CS17. On this occasion, I thought it appropriate to share the course design rationale with the readers of *Conduit*.

Multiple programming languages

One of our mottoes during the initial design of CS17–CS18 was "beyond syntax." This reflects our goal that students understand aspects of program-

Philip Klein

The infamous Devil Cat that represents the worst case opponent a computer algorithm can be up against in CS17–CS18.

"I would be delighted to hear from alumni with thoughts on introductory computer science education."

ming and programming languages in a way that transcends the syntax of a particular programming language. To be able to see things from such a "meta" perspective, students need to learn several programming languages. Once they do, they are in a better position to understand what is essential to expressing computations and what is idiosyncratic.

Multiple programming paradigms

For a similar reason, students need to learn more than one programming paradigm. Examples of programming paradigms are procedural programming, functional programming, and object-oriented programming. Programming paradigms have evolved over the years, and no one paradigm is perfect for all programming problems. Even if one is determined to use object-oriented programming, say, experience with multiple paradigms is essential to understanding the motivation for programming-language features that support object-oriented programming. Moreover, ideas from functional programming are seeping into the mainstream, e.g., the MapReduce programming model used in Google.

Mix together the learning of programming with the learning of data structures and algorithm analysis

The traditional approach to the introductory computer science curriculum is to have a course in programming followed by a course in data structures and algorithms analysis. We believe that these topics should be taught together in an integrated fashion. First, computational performance (how much time or memory a computation requires) is critically important to at least some programs and applications. Second, the two topics enliven and enrich each other. Data structures and algorithms provide great examples of programming schemata, and the need for greater efficiency can motivate the need for more sophisticated approaches to a problem. Third, learning to analyze the efficiency of a program is just a part of learning to understand a program more precisely. More precise understanding of programs leads to faster debugging and to better (sometimes simpler) designs.

Assignments at a variety of scales

In CS17–CS18, there are programming assignments at a variety of scales: quizzes, labs, homeworks, and projects. A quiz takes five minutes, a lab takes an hour, a homework can take several hours, and a project can take a day (spread out over several weeks). Part of learning to program quickly and effectively is learning to write small programs very

quickly; confidently writing small programs is the first skill needed to write larger ones.

Multiple application areas

We draw our examples and assignments from a wide variety of areas of computer science, including natural-language processing, programming languages, databases, computational biology, numerical processing, simulation, financial computation, and graphics and graphical interfaces. Our students discover the pleasure to be had from writing (from scratch) a program that can hold a (very rudimentary) conversation with a person, a program that can play a passable game of mancala, and a program that implements a programming language. Most important, our students experience, again and again, the intellectual satisfaction of finding an elegant solution to a knotty computational problem. Our belief is that the development of that skill is the best foundation for continued study in all areas of computer science.

Many Brown CS courses provide already-written programs for students to build on in doing their assignments. Certainly a student can obtain more impressive results if she builds on previously written programs. However, we feel it is both more satisfying and more educational for programmers to write a program from scratch.

For these reasons, very little support code is used in CS17–CS18. On the rare occasions when we provide code to the students to help them with assignments, the amount of code is small and the students are expected to understand all of it. Even the use of libraries (programs provided with a programming language) is minimized. The students themselves (sometimes collaboratively) write the programs themselves, and they are expected to completely understand the programs they write. This practice is consistent with the most important CS17–CS18 motto: "No magic." This motto reflects our goal of making everything transparent to as great a degree as possible. The questions "how?" and "why?" are always welcome in CS17–CS18.

Conclusion

Involvement with CS17–CS18 has been a passion of mine for much of the past decade. It's given me a renewed appreciation for our students' talent, dedication, and enthusiasm for learning. I'm very grateful to the many students who served as TAs for CS17–CS18 and helped to shape the course. I would be delighted to hear from alumni with thoughts on introductory computer science education. **C!**

Autonomous 🍴 ⛴ Bidding 🚗 🥾 Agents: Strategies and Lessons from TAC Travel 🧳 🚲

TRADE IS A QUINTESSENTIAL HUMAN ACTIVITY.

Throughout history, the institutions and artifacts of trade, such as markets and currency, have evolved hand in hand with major technological advances, such as the printing press and telecommunication networks. The rise of the Internet in the past decade is another transformative advance. In the new online markets, buyers and sellers have unprecedented opportunities for trade through a wide array of novel mechanisms. For example, eBay, the largest online auction site, is responsible for creating viable markets for numerous idiosyncratic goods, and providing a medium where thousands of small businesses earn their livelihood.

Compared to other interactive decision-making domains (e.g., politics), the trading task is particularly amenable to automation. Successfully automating trading, however, depends not only on trusted, reliable software interfaces, but also on being able to render effective decisions about which trading actions to take (e.g., what to exchange, when, and at what price). The topic of effective decision-making is paramount to many disciplines, including psychology, economics, operations research, philosophy, and computer science. The field of artificial intelligence draws on principles from these subjects and others to develop techniques for decision-making by "autonomous agents."

Autonomous agents are software programs that make decisions without direct human intervention. Even though computer software is conceived of by (human) designers and implemented by (human) programmers, we refer to programs as autonomous to the extent that their decisions are not rote implementations of instructions spelled out by a human. For example, a program that carries out the direction "Bid $99 for eBay item 123" would not be considered an autonomous trading agent. But a program that submits this bid based on the direction "Buy a digital camera at a good price" could be credibly attributed a meaningful degree of autonomy.

1 THE TRADING AGENT COMPETITION

The annual Trading Agent Competition (TAC) challenges its entrants to design autonomous agents capable of effective online trading. The first TAC, held in July 2000 in Boston, attracted sixteen entrants from six countries in North America, Europe, and Asia to play a challenging market game in the domain of travel shopping. Excitement generated from this event led to refinement of the game rules and continuation of regular tournaments with increasing levels of competition over the next six years. Year-by-year entrants improved their designs, developing new ideas and building on successful techniques in previous tournaments. Published accounts of TAC agents and TAC-based experimental studies now amount to a substantial literature [1].

TAC dresses in a familiar narrative several realistic features of trading in electronic markets. A TAC agent's objective is to procure travel packages—comprised of airline flights, hotel reservations, and entertainment events—as inexpensively as possible, while satisfying its clients' preferences to the greatest extent possible. Shopping for travel goods is a routine activity for many people, and assembling trips for clients is a plausible application for agent technology.

But the key feature captured by the travel domain, and hence TAC, is that goods are highly interdependent (e.g.,

"Shopping for travel goods is a routine activity for many people, and assembling trips for clients is a plausible application for agent technology."

flight and hotel reservations must be coordinated), yet the markets for these goods operate independently. A second important feature of TAC is that the three kinds of goods are traded in different kinds of markets, each of which presents distinct challenges. Participants must grapple with all three while constructing their agent strategies.

Flights Flights are sold by TACAir in dynamic posted-pricing environments. The hallmark of a posted-price mechanism is that a designated party sets a price that the other parties can "take or leave." However, posted-price markets may differ regarding limits on quantity, the manner or frequency by which prices change, or other features. In the case of TAC flight markets, TACAir posts prices for an effectively unlimited quantity of flights to and from TAC-Town on each day, and TAC agents can buy any number of flights by submitting bids at or above those prices. TACAir updates prices according to a known stochastic process. This process has both a revealed and hidden state but is not affected in any way by the TAC agents' actions.

The fundamental issue regarding TAC flight decisions is a common one: balancing concern about future price increases with the benefit of delaying commitment to travel on particular days. If flight prices were nonincreasing, agents would simply delay their flight purchases until the end of the game, when all uncertainty about hotel markets has been resolved. By committing to a flight any earlier, an agent risks finding that its choice was suboptimal, based on subsequent shifts in hotel prices or availability. An extreme (but not unusual) instance of this risk is that it may end up wasting the flight entirely, if it cannot obtain hotel rooms to compile a feasible trip on the corresponding days.

Hotels In a *simultaneous ascending auction* (SimAA) [2], goods are sold to agents through an array of ascending auctions, one for each good. The auctions proceed concurrently, and bidding is organized in rounds. At any given time, the price quote is defined to be the highest bid received thus far, or zero if there are no bids as of yet. The ask price is the price quote plus a fixed increment. To be admissible, a new bid must "beat the quote" by offering at least the ask price. If an auction receives multiple admissible bids in a given round, it admits the highest (breaking ties arbitrarily). An auction is quiescent when a round passes with no new admissible bids. When all are simulta-

neously quiescent, the auctions close and their respective goods are allotted as per the last admitted bids.

The TAC hotel auctions run simultaneously but differ from the abstract SimAA mechanism in two basic ways. First, each TAC hotel auction is multi-unit: the top k bidders are allocated the k units, and the clearing price is the lowest winning bid. Second, rather than wait until all auctions are quiescent, one randomly selected auction closes each minute. In this respect TAC hotel auctions represent a hybrid between simultaneous and sequential auctions. Still, TAC hotel auctions and SimAAs share many of the same characteristics. Most importantly, because no transaction in a SimAA is contracted until all are, an agent's bidding strategy in one auction cannot be contingent on the outcome in another. Similarly, an agent bidding for a set of hotels on contiguous days runs the risk that it will win some but not all hotels it desires.

Entertainment Most TAC designers treat entertainment trading as a task only loosely coupled to flight and hotel bidding. The markets are clearly interdependent, as the value of an entertainment ticket depends on what other tickets the agent holds, as well as the possible travel packages it can assemble for its clients. The relationship is relatively weak, however, since entertainment merely provides bonus utility; it does not affect trip feasibility like flights and hotels (e.g., a trip is infeasible if it includes only flights but no hotel reservations). Often a ticket not used for one client can be given to another or sold to another agent. Entertainment markets are open throughout the game, and are not subject to time-dependent price movements (like flight markets) or rigid clearing schedules (like hotel markets).

2 RoxyBot, Brown's TAC Entrant

Since TAC's inception, Brown has entered successive modifications of its autonomous trading agent, RoxyBot, developed by my research group.[1] In our approach to the problem of bidding on interdependent goods in the separate TAC markets, we adopted some simplifying assumptions. Rather than tackle the game-theoretic problem of characterizing strategic equilibria, we focused on a single agent's (decision-theoretic) problem of optimizing its own bidding behavior, assuming the other agents' strategies are fixed. In addition, we assumed that the

environment can be modeled in terms of the agent's predictions about market clearing prices. These prices serve to summarize the relevant information hidden in other agents' bidding strategies. These two assumptions—fixed other-agent behaviors and market information encapsulated by prices—support RoxyBot's modular design, which consists of two key steps: *price prediction* and *optimization.*

In 2000, RoxyBot predicted prices in the form of point estimates and then solved the ensuing deterministic optimization problem. This strategy proved its effectiveness, as RoxyBot was one of the two top-scoring agents in that tournament [3]. Nonetheless, RoxyBot-00 was hindered by the fact that it could not explicitly reason about variance within prices. In the years since 2000, we recast the key challenges faced by TAC agents as several different stochastic bidding problems, whose solutions necessarily exploit price predictions in the form of distributions. In spite of our perseverance, RoxyBot fared unimpressively in tournament conditions year after year . . . until 2006. Half a decade in the laboratory spent searching for bidding heuristics that can exploit stochastic information at reasonable computational expense finally bore fruit, as RoxyBot emerged victorious in TAC-06. The "secret" of RoxyBot-06's success is, in a nutshell: hotel price prediction by simulating simultaneous ascending auctions, and bidding based on the sample average approximation method.

3 2006 Tournament Results

Table 1 lists the agents entered in TAC-06 and summarizes the outcome. RoxyBot dominated the seeding round, which consisted of 960 games. The finals comprised 165 games over three days, with the 80 games on the last day weighted 1.5 times as much as the 85 over the first two days. In spite of its glowing performance in the preliminary (qualifying and seeding) rounds on the first day of the finals, RoxyBot finished third, behind Mertacor and Walverine—the top scorers in 2005. As it happens, RoxyBot's optimization routine, which was designed for stochastic hotel and entertainment price predictions, was accidentally fed deterministic predictions (i.e., point price estimates) for entertainment. Moreover, these predictions were fixed, rather than adapted based on recent game history.

On days two and three, RoxyBot ran properly, basing its bidding in all auctions on recent stochastic information. Moreover, the agent was upgraded after day one to bid on flights not just once, but twice, during each minute. This enabled the agent to delay its bidding somewhat at the end of a game for flights whose prices are decreasing. No doubt this minor modification enabled RoxyBot to emerge victorious in 2006, edging out Walverine by a whisker, below the integer precision reported in Table 1. The actual margin was 0.22—a mere 22 parts in 400,000. Adjusting for control variates [4] spreads the top two finishers a bit further. Accounting for RoxyBot's difficulties on day one of the finals, the difference between the bidding capabilities of the first and second place agents is perhaps not as close as it seems.

4 Lessons Learned

In the seven years since its introduction in 2000, my research group has put a tremendous amount of effort into organizing and competing in TAC. Is this effort justified?

Not surprisingly, we would argue that the attention and energy focused on TAC by the trading agent research community has been worthwhile. As a result of this endeavor, we have at our disposal a bonanza of new

Agent	Affiliation	Seeding	Finals	Adjustment Factor
RoxyBot	Brown U	4148	4032	−5
Walverine	U Michigan	3992	4032	−17
WhiteDolphin	U Southampton	3901	3936	−2
006	Swedish Inst Comp Sci	3882	3902	−27
Mertacor	Aristotle U Thessaloniki	3509	3880	−16
L-Agent	Carnegie Mellon U	3284	3860	7
kin agent	U Macau	3897	3725	0
UTTA	U Tehran	1726	2680	−14

Table 1: TAC-06 scores, seeding and final rounds, with adjustment factors based on control variates.

trading agent techniques: bidding heuristics, price-prediction methods, learning algorithms, and optimization models. These techniques have all been vetted in a challenging and competitive benchmark environment, refined, and compared with alternatives. Experimental analysis of TAC strategies has produced a treasure trove of data, resolving empirical questions of efficacy and pointing trading agent researchers in promising new directions.

Even more significant than the individual techniques, in our view, are the broader lessons we can extract from the TAC experience. One might have predicted that the competition would converge on a single best-practice solution, with slight variations, or would devolve into a contest among incomparable ad hoc programs. Instead, what we observe is the emergence of a common (if not universal) trading agent architecture, with an interesting diversity of competing approaches explored for the various subproblems. Specific TAC techniques have evolved along distinct threads through generations of the tournament, but—aided by common functional structure—with substantial cross-fertilization across entrants.

Thus, we consider the body of techniques distilled from trading agents competing in TAC to provide a firm engineering foundation for autonomous bidding in real-world domains. Continued research and practice in TAC Travel, TAC Supply Chain Management (ongoing since 2003), and TAC Market Design (debuting in 2007), and other trading environments will refine, validate, and likely supersede our current body of knowledge, ultimately rendering more accessible and routine the art of trading agent design. **C!**

References

[1] http://tac.eecs.umich.edu/researchreport.html.

[2] Peter Cramton. Simultaneous ascending auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions.* MIT Press, 2006.

[3] Amy Greenwald. Internet agent economics: A trading agent competition. *Conduit,* 9(2):1–4, 2000.

[4] Sheldon M. Ross. *Simulation.* Academic Press, third edition, 2002.
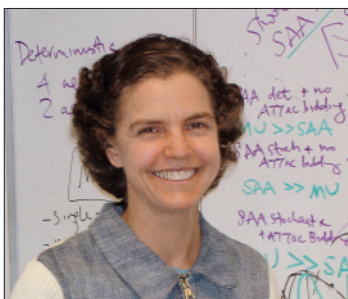
Amy Greenwald designs and implements AI agents that interact effectively in multiagent environments. She also works on developing theories that can explain, and ultimately predict, the dynamics of such interactions.

Ph.D. Candidate Danfeng Yao Receives Best Student Paper Award at ICICS

Ph.D. candidate Danfeng Yao received the Best Student Paper Award at the Eighth International Conference on Information and Communications Security (ICICS '06) for her paper titled, "Point-Based Trust: Define How Much Privacy Is Worth." The work was in collaboration with her advisor, Roberto Tamassia, Keith Frikken at Miami University, and Mikhail Atallah at Purdue University. The authors were also given a cash prize for their contributions. The paper was presented as the first talk on the second day of the conference.

The paper by Danfeng and her colleagues addresses the privacy protection problem on the web and proposes a solution to protect sensitive personal information. The ultimate goal of their work is to make the Internet a safer environment for conducting digital transactions such as online shopping, e-banking, and e-commerce in general.

The authors propose a point-based authorization model, which supports personalized access policies. In this model, a consumer can choose which private information to disclose in an online transaction. For example, if a consumer considers their phone number sensitive, they can then choose to disclose an e-mail address as the contact information. The paper gives a cryptographic protocol that guarantees the consumer's choice is always optimal and minimizes privacy loss.

The point-based model makes it possible to compensate consumers disclosing their private information, which creates incentives for consumers to actively participate in on-line transactions.

For more information about this work, please visit http://www.cs.brown.edu/people/dyao/-icics-award.html.

Faculty Notes

Michael J. Black

Michael was on sabbatical in the fall meaning that he was able to travel even more than usual. He co-organized an NSF-funded workshop in Montevideo, Uruguay, on Vision by Brains and Machines, which brought researchers working on both human and machine vision together. In November he gave a public lecture in Athens, hosted by the National Hellenic Research Foundation, on "Building the bionic body: Restoring movement to the severely disabled with a brain-machine interface."

He also traveled to India, where he gave an invited talk at the Indian Conference on Computer Vision Graphics and Image Processing (see photo below). After the conference he visited Microsoft Research India in Bangalore, where his former thesis advisor is the managing director.

Along with Chad Jenkins and John Donoghue in Neuroscience, Michael received a $314,880 infrastructure grant from the Office of Naval Research to support the purchase of motion capture equipment and a robotic hand.

Ugur Cetintemel

In January, Ugur co-organized an NSF-sponsored workshop on Data Management for Mobile Sensor Networks (Mobi-Sensors '07) in Pittsburgh. The workshop brought together researchers and practitioners from academia and industry to discuss an emerging class of ubiquitous, sensor-driven applications and their needs with respect to data management.

In the spring semester, Ugur started teaching CS18, the second of the CS17/CS18 introductory course sequence. The course integrates object-oriented programming concepts with algorithms, data structures and their analysis.

John F. Hughes

Spike has two students, Tomer Moscovich and Olga Karpenko, finishing up this year. Tomer's already done and off to a postdoc at the University of Toronto, and Olga's writing her thesis. He continues to slog his way through writing the new edition of *Computer Graphics: Principles and Practice*, while spending more and more time reading machine-learning papers.

Sorin Istrail

Sorin presented a keynote lecture, "Randomness is Beautiful: In Search of von Neumann" at the 2006 NIH Graduate Student Symposium, an invited lecture at the National Institute of Standards and Technology, and gave a series of lectures, "Logic Functions of the Genomic cis-Regulatory Code," at the Lipari Island School on Computational Biology. He also was an invited member of the NSF panel on designing the next program initiative for petaflop-scale supercomputing and the NIH panel on Biodata Management and Analysis. He coauthored two papers in Science, "The Genome of the Sea Urchin *Strongylocentrotus purpuratus*" and "The Transcriptome of the Sea Urchin Embryo," and was coeditor of a special issue of *Discrete Applied Mathematics* on computational molecular biology.

Two additions were made to his group: a postdoctoral researcher, Fumei Lam, working on computational methods for SNPs and haplotype analysis, and a research assistant professor from "Al. I. Cuza" University, Iasi, Romania, Gabriel Negara, working on lattice models of protein folding.

Last fall, Sorin was awarded an endowed chair and became the Julie Nguyen Brown Professor of Computational and Mathematical Sciences. He was also appointed Director of the Center for Computational Molecular Biology (CCMB) in September 2006. In December he organized the IPP/CCMB Symposium, "The Genome and the Computational Sciences: The Next Paradigms," featuring thirteen CCMB Inaugural Distinguished Lectures given by some of the most distinguished scientists of the genome and biotechnology era. He started new research collaborations with medical faculty from BrownMed on genome-wide disease associations for sudden cardiac death, and risk for substance abuse by children with *in utero* cocaine exposure. In addition, seed funding from the Office of the Vice President for Research supports a collaboration with scientists from Oak Ridge National Laboratory on computational reconstruction of protein folding from neutron scattering data.

Odest Chadwicke Jenkins

Chad and Sarah Jenkins welcomed their second child, Wesley Spencer Jenkins, to the world in early February. Chad has also begun a collaboration with iRobot Corporation, funded by DARPA, to improve human-robot interaction for collaborative teams of humans and heterogeneous robots.

David Laidlaw

David, RISD colleague Fritz Drury, and evolutionary biologist Sharon Swartz taught the third version of their joint Brown/RISD course "Virtual Reality Design for Science." Not only were the faculty interdisciplinary: the class has students from both institutions. Help from CCV staff, Andy Forsberg, and grad students Dan Keefe and Daniel Acevedo was invaluable. Students mocked up some great interfaces for scientists to consider using to study the motion of bats in flight and the interactions those motions create with the surrounding fluid.

In another collaborative effort, David was a Co-PI on a recent $1.8 million grant from the Keck Foundation to develop hardware and software tools for studying the motion of many different animals. The hardware will use dual fluoroscopes to capture high-speed, high-resolution x-ray movies from two directions, so that 3-D motion can be calculated.

David is also teaching CS190 this semester and welcomes alumni to visit class and give their perspective on the real world of software engineering. Recent visitors to class include Sam Cates, Catherine Hill, and Andy Simon.

The Visualization Research Lab is proud to report that Song Zhang has finished his Ph.D., and that Dan Keefe and Liz Marai are about to. We also welcome new postdoctoral researcher Jian Chen.

On the new hardware front, Andy van Dam's group, David's group, and CCV have worked together to build a 3x3 projector high-resolution display system. It is close to coming online, where it will be used to investigate the value of higher resolution and brightness than the current virtual reality Cave display system can deliver.

Anna Lysyanskaya

I am back from my sabbatical, which I spent visiting the Weizmann Institute of Science (Rehovot, Israel—see photo with Mira Meyerovich) and Institute for Pure

and Applied Mathematics (UCLA, Los Angeles). The latter, in particular, was a great experience: it was a special semester-long program in cryptography. At the moment I am buried very deeply underneath a million papers that I have to review for various conferences: I am serving on too many program committees this year. Also as I am typing this, I am about to rush to the airport—I seem to be going somewhere every week this day: it's always either a talk, or a conference, or a program committee meeting. But when I am not reviewing all those papers and not dashing off to the airport, I manage to get some research done, too! I am very proud of my Ph.D. students, Melissa Chase and Mira Meyerovich; both of them will graduate next year.

Claire Mathieu

Together with her student Aparna Das and other colleagues, Claire Mathieu has worked on designing and analyzing bidding strategies for keyword auctions such as Google's Adwords. They proposed a strategy for repeated bidding that converges to a "satisfactory" equilibrium, and the result will be presented at the Electronic Commerce conference.

During the winter break, after vacationing in France, Claire attended a conference in New Orleans, where she ate good food and enjoyed the historic district, but also saw water stains on walls of houses, heard about muggings and mice infestations, and personally witnessed a big dark mouse in her hotel bathroom! To her, going to New Orleans felt like visiting a Third World country.

Barbara Meier

I am teaching a new course this semester, CS128 Intermediate Computer Animation. We are concentrating on learning advanced modeling, rigging, animation, shading, and lighting techniques, and how to incorporate these into effective workflows. This is especially challenging because these workflows can change dramatically even from year to year with new underlying algorithms, more control made possible by faster processing times, and advances in user interfaces. Fortunately, the underlying principles of good animation remain constant! I welcome any former Brown students and my former colleagues from the animation industry to guest lecture in CS128 next spring or in my introductory animation course in the fall. Tales from the front lines keep us abreast of how new technologies are folded into existing pipelines and are also enormously helpful in reinforcing that no technology can hide a bad idea or boring story.

Ben Raphael

Ben joined the department in the fall and is still figuring out how faculty balance the various demands of the job. This spring he is teaching a graduate computational biology course called "Genomes, Networks, and Cancer." He co-authored papers that recently appeared in the Proceedings of the National Academy of Sciences, the Research in Computational Molecular Biology (RECOMB) conference, and PLOS Biology. Finally, he was featured as one of "Tomorrow's PIs" (principal investigators) in the December issue of *Genome Technology* magazine. Someday, he plans to finish unpacking the boxes in his office...

Ph.D. Profiles

Nesime Tatbul

Nesime Tatbul is one of our recent Ph.D. graduates. She defended her thesis in November 2006, shortly before joining the Department of Computer Science at ETH Zurich in Switzerland as a tenure-track assistant professor.

Nesime joined our department in 1999 and has since been a member of the Brown Data Management Group. Her advisor was Professor Stan Zdonik. As part of the Aurora/Borealis project team, she was involved in the design and implementation of a large-scale data stream processing system, which later has been turned into a successful start-up company called StreamBase Systems. Her Ph.D. research has focused on high-performance processing on real-time data streams such as stock data feeds from financial markets. Her dissertation, entitled "Load Shedding Techniques for Data Stream Management Systems," showed how large numbers of continuous queries (e.g., "Report when hourly average price of a stock gets above $20") can be processed with minimal latency in the face of high data arrival rates (e.g., 100,000+ data items per second). In addition to her dissertation work, Nesime also spent a summer at the IBM Almaden Research Center working on a database caching project, and served as a part-time consultant for the U.S. Army Research Institute of Environmental Medicine working on data management techniques for personal-area wireless sensor networks.

She is currently enjoying her new life as a faculty member in one of the best cities to live in the world, and is looking forward to visiting Brown in May for the commencement.

Song Zhang

Song Zhang received his undergraduate degree from Nankai University in his hometown, Tianjin, China. In 1998 he entered the Ph.D. program at Brown in Computer Science and began to work with his advisor, Professor David H. Laidlaw, in visualization and computer graphics. His thesis work focuses on modeling, analyzing and visualizing human brain structures from diffusion imaging data. He defended his Ph.D. thesis, "Revealing White Matter Fiber Structure with Diffusion Imaging," in July 2006. Since August 2006, he has been a tenure-track assistant professor in the Computer Science and Engineering Department of Mississippi State University. He is currently affiliated with the Institute for Neurocognitive Science and Technology, the Center for Computational Sciences, and the GeoResources Institute. He is collaborating with professors and students all across campus on visualization projects that intrigue him.

Tomer Moscovich

Tomer Moscovich defended his thesis on "Principles and Applications of Multi-touch Interaction" in December, just in time to move out of his apartment, find a new place in Toronto, and move there to start a post-doc with Ravin Balakrishnan's group at the University of Toronto at the end of January **C!**

Faculty Notes

John Savage

John Savage was scheduled to give a talk at the International Conference on Computer Aided Design (ICCAD) in November 2006 on "Nanowire Addressing with Randomized Contact Decoders" as well as to honor Franco Preparata by giving the opening address at FrancoFest. Unfortunately, illness intervened and substitute speakers were found for both talks. He is back in good health and active in research and teaching. He taught CS257, Introduction to Nanocomputing, in the fall and is teaching CS256, Applied Theory of Computation, in the spring. He continues to serve as chair of the departmental Curriculum Committee and is a member of the Working Group on Global Science and Technology which is contributing to the Provost's study on internationalization. He was a member of the Steering and Program Committees of HPC Nano '06 Workshop at Supercomputing 2006.

Meinolf Sellmann

Last year was probably the most productive research year that I have had in my career so far. Two results stand out, one theoretical and one practical. The first was to show that context-free grammars can be used to efficiently constrain the space of feasible solutions. In particular, I showed that the Cocke-Younger-Kasami parser can be adapted to achieve generalized arc-consistency for context-free grammar constraints in polynomial time.

The second main contribution was achieved in collaboration with Carlos Ansotegui from the University of Lleida (Spain). We developed of a novel method for learning favorable value heuristics in restarted systematic solvers. On various application domains, we were able to show speed-ups of one or more orders of magnitude.

In terms of undergraduate research, I am very happy that two of my 2006 papers, which were presented at the leading conferences with respect to constraint programming and CP/OR hybridization, respectively, resulted from my collaboration with undergraduate students Dan Heller and Dan Leventhal (who received an honorable mention from the CRA).

Pascal Van Hentenryck

The MIT Press published Pascal and Russell Bent's book *Online Stochastic Combinatorial Optimization* in October 2006. The Program Committee of IJCAI '07 gave a distinguished paper award to Luc Mercier and Pascal for their paper "Performance Analysis of Online Anticipatory Algorithms for Large Multistage Stochastic Programs." There were three distinguished paper awards out of 1353 submissions. Russell and Pascal also published their paper on "Waiting and Relocation Strategies in Online Stochastic Vehicle Routing" at the same conference. Laurent Michel, one of Pascal's former Ph.D. students, was awarded a CAREER Award for his work on constraint programming in early 2007. Ionut Aron and Yannis Vergados successfully defended their Ph.D. thesis in January 2007 and December 2006.

Stan Zdonik

Stan Zdonik was inducted this year as an ACM Fellow for his contributions to database systems and data management. He also co-chaired two NSF Workshops, one in November 2006 on the Role of Data Management in the new NSF GENI Initiative and another in January 2007 on Data Management for Mobile Sensor Networks. He was on sabbatical at MIT for the 2006 calendar year where he worked with collaborators on C-Store, a column-oriented database management system. C-store can process read-mostly workloads orders of magnitude faster than its row-oriented counterparts. **C!**

Professor Sellmann Awarded NSF Career Award for his Cornflower Project

The Objective of the Cornflower Project The dawn of the new century casts light on three dramatic economic challenges that will determine our future as a society: demography, globalization, and shortage of natural resources. Today we need to develop the technology that will allow our children to maintain our standard of living. Therefore, we need to find ways to increase our economical efficiency. Computer science can play a decisive role when facing this challenge. After more than sixty years of research, algorithmic computer science can offer a lot to help saving. However, the main impact of computational optimization support is currently limited to large companies in few core application areas such as the transportation industry. Specialized algorithmic solutions have been shown to be extremely successful in these domains. While the algorithmic technologies that were developed are by no means specific to the current areas of application, the main obstacle for a broader realization of their vast potential is largely due to their lacking ease of use. Therefore, the main goal of the Cornflower Project is to develop techniques that allow inexperienced users to exploit optimization power efficiently.

The Intellectual Challenge Optimization techniques have been developed in three historically separate research areas: constraint programming, operations research, and algorithm theory. While the main focus of the project is to provide a high level of automization and algorithms that can be hooked to intuitive modeling primitives that facilitate the use of intelligent optimization support, we do not stop at the boundaries of traditional research areas. Instead, we integrate and hybridize ideas developed in different communities in order to provide easily accessible high performance optimization technology.

Particularly, we focus on the development of high-level constraints that allow users to model their problems as conjunctions of intuitive substructures and provide hybrid methods for their efficient combination. Moreover, we develop automization techniques for the handling of symmetries that can be the cause of severe inefficiencies when handled poorly.

The Expected Impact The main goal of the Cornflower project is to solve the algorithmic problems that arise in the context of optimization driven decision support systems that are intuitive to use and that provide a high level of automization. By making algorithms and methods publicly available in the Cornflower Library, the project contributes to widening the access to computational decision support.

Our goal to broaden the accessibility of computational decision support while breaking the barriers between theoretical and practical computer science goes hand in hand with the educational goals of the project. Optimization techniques play an ever more important role for many other CS disciplines. By developing optimization courses that integrate methods from various research areas, Cornflower helps to educate the next generation of experts in combinatorial algorithms, while students moving on to other disciplines take away a sound understanding of how to tackle combinatorial problems and how to utilize standard solvers for their own purposes. Attracting both students who are mainly interested in technical methodology as well as those who are primarily attracted to specific applications is also a promising approach to broadening participation in science at large. Through the outreach program Artemis, a five-week summer program for high-school girls, young women are encouraged to pursue careers in science and engineering.

Faculty Awards and Honors

Professor Sorin Istrail Receives OVPR Seed Fund Award

Sorin Istrail, the Julie Nguyen Brown Professor of Computational and Mathematical Sciences and Professor of Computer Science and Director of the Center for Computational Molecular Biology, has received a $65,000 targeted research seed fund award for scientific computing. This award was bestowed by the Office of the Vice President for Research and funded by the Office of the Provost.

Istrail will use the seed money to begin a groundbreaking project: Computationally reconstruct a single protein molecule folding in vivo. Istrail says meeting this goal would go a long way toward advancing the science of genomics and improving human health.

"Proteins are the work horses of the cell," he said. "How they form and shift shape is one of the major secrets of biology—and also a major key to understanding how proteins help with metabolism, immune response, and cell signaling. And many diseases, such as Alzheimer's and certain kinds of cancer, can be traced to protein misfolding. We want to try to understand how this process happens. To do that, you have to be bold."

Istrail will collaborate with scientists at Oak Ridge National Laboratory and the California Institute of Technology to develop new computational environments for molecular reconstruction.

Professors Preparata and Tamassia Among Most Highly Cited Computer Scientists

The Institute for Scientific Information (ISI), now Thomson Scientific, has included Franco Preparata and Roberto Tamassia in the list of the most highly cited computer science authors worldwide, based on citations in venues indexed by ISI for the period 1981–1999. In the website ISIHighlyCited.com, ISI features the "preeminent individual researchers in each of 21 subject categories who have demonstrated great influence in their field as measured by citations to their work—the intellectual debt acknowledged by their colleagues." For the field of computer science, the list contains 319 people. In addition to Brown CS faculty Preparata and Tamassia, the list includes Brown CS alums Ed Lazowska and John Stankovic. The venues indexed by ISI

include the most prestigious scientific journals and selected conference proceedings. Articles published in such venues are the primary source used by the National Research Council in assessing scholarly productivity in science and engineering.

Professor Stan Zdonik Named ACM Fellow

Professor of Computer Science Stan Zdonik was named a 2006 ACM Fellow for his contributions to data management and database systems.
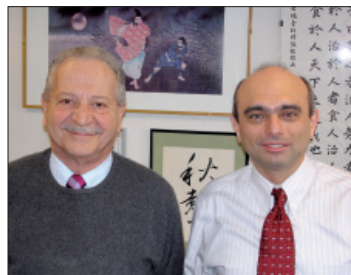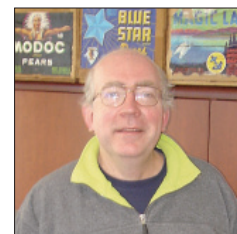
The award honors contributions to both the practical and theoretical aspects of computing and information technology. For 2006, forty-one ACM members were awarded the designation of Fellow with nine being members of the database community.

Said ACM President Stuart Feldman, "Their work reflects outstanding displays of creativity and commitment to the computing community, which continues to drive innovation in industries and enterprises across the globe. These individuals deserve our acclaim for providing dedicated leadership, solving complex problems, and pursuing productive careers in information technology that have advanced the quality of life for people everywhere."

The ACM Fellows Program was established in 1993 to recognize and honor outstanding ACM members for their achievements in computer science and information technology and for their significant contributions to the mission of the ACM. The ACM Fellows serve as distinguished colleagues to whom the ACM and its members look for guidance and leadership as the world of information technology evolves.

Herlihy Awarded Microsoft Gift for Research in Software Transactional Memory

Professor Maurice Herlihy was awarded a $75,000 Microsoft gift for research in Software Transactional Memory. Herlihy will use the gift to continue development of SXM, a C# software transac-

tional memory package he developed while on sabbatical at Micro-soft Research Cambridge. Transactional memory is an alternative computational model in which threads synchronize by optimistic transactions, which promise to alleviate many of the problems associated with locking. SXM currently supports atomic object factories that allow users their own run-time synchronization mechanisms. Herlihy plans to revisit this design to make it more accessible (users will define new factories in C# instead of MSIL) and to add new features to support a range of recently-proposed STM architectures and algorithms.

Herlihy will use the new SXM2 platform and will release the source under license so that others may do the same.

Eli Upfal Receives Yahoo! Research Alliance Gift

Professor Eli Upfal and Michael Mitzenmacher, professor of computer science at Harvard, have been awarded a $100,000 Yahoo! Research Alliance gift for their research in Algorithms and Modeling for Large-Scale Web Applications.

The funding will assist in the development of a theoretically well-founded framework for the design and analysis of algorithms for large-scale Web-related applications including: efficient sampling and analysis for content matching; and modeling and analyzing the dynamic structure of Internet-based social networks and communication within these networks.

Pascal Van Hentenryck Receives the 2006 ACP Award

Pascal received the 2006 ACP Award for Research Excellence in Constraint Programming in Nantes during the 12th International Conference on Principles and Practice of Constraint Programming in September 2006. He is the second recipient of this award which was initiated in 2005. The award citation mentions Pascal's "pioneering work in Constraint Programming, his numerous and influential results in algorithm and language design, and his remarkable scientific and technological impact in the optimization field." **C!**

Pascal Van Hentenryck (left) receives the ACP award for research excellence in constraint programming.

Daniel Leventhal and Leo Meyerovich

Daniel Leventhal and Leo Meyerovich Receive Honorable Mentions in CRA's 2007 Outstanding Undergraduate Awards

Undergraduate students Daniel Leventhal and Leo Meyerovich have received Honorable Mention awards from the Computing Research Association (CRA). CRA's Outstanding Undergraduate Awards program recognizes undergraduate students who show outstanding research potential in an area of computing research. The award committee looks for demonstrated excellence of computing research ability and also considers the student's academic record and service to the community.

The awards, supported by Microsoft Research and Mitsubishi Electric Research Labs (MERL), are presented at one of the major computing research conferences sponsored by CRA, ACM, the IEEE Computer Society, SIAM, AAAI, or USENIX.
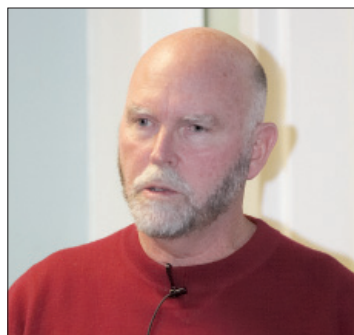
For every year since 1998, at least one student from Brown's Department of Computer Science has received an Outstanding Undergraduate Award.

Congratulations Dan & Leo!

37th IPP Symposium: "The Genome and the Computational Sciences: The Next Paradigms"

Organized Jointly with the Center for Computational Molecular Biology

Sorin Istrail

Biotech entrepreneur J. Craig Venter delivered a special plenary lecture at the 37th Computer Science Department Industrial Partners Program Symposium held December 6-8, 2006. The symposium, titled "The Genome and the Computational Sciences: The Next Paradigms," brought together thirteen leading thinkers from academia, business and government to discuss new ways to mine the wealth of data contained in the genomes of microbes, plans and animals. With the goal of better understanding human development and evolution, disease and animals – as well as creating new medical tests and treatments and even clean fuels and pest-resistant crops – computational biology is booming, as evidenced by a burst of new research programs and new startup companies.

Dr. Venter, founder of Celera Genomics, president of the J. Craig Venter Institute and co-founder of Synthetic Genomics, spoke about new ways to use genomic data to improve everything from medicine to the environment. Other speakers were David Altshuler of Harvard Medical School and the Broad Institute of MIT and Harvard, who discussed the role of affordable genome-wide genotyping technologies in whole-genome disease association studies; David Barker, speaking about the integration of genomic, epigenetic, and gene expression data by computational tools to reveal underlying mechanisms; Brown's Leon Cooper talked about "Is theory possible in neurosciences?"; Stephen Hoffman, founder and CEO/CSO of Sanaria and co-founder and chairman of Protein Potential, both Maryland-based vaccine development companies, described the journey from genomics molecular immunology and DNA vaccines to an attenuated whole-parasite malaria vaccine; Christopher Johnson of the University of Utah discussed the role of biological imaging and visualization in building and testing biological models of increasingly complex phenomena; and Jonathan King of MIT spoke about protein folding, in particular elucidating the sequence control of beta-sheet in proteins. IBM's Eric Kronstadt projected the huge computational power expected by the end of the next decade and described some exciting biological/genomic applications. Pavel Pevsner, UCSD, surveyed the state of play in theories of chromosomal evolution and rearrangements; David E. Shaw of D.E. Shaw Research described the state of the art in biomolecular simulation and explored the potential role of high-performance computing

technologies in extending current capabilities; Jeffrey Skolnick of Georgia Tech discussed a novel method for predicting protein structure, function and druggability based on the sequence-to-structure-to-function paradigm; Jeremy Smith of Oak Ridge National Labs, gave examples of the use of computer simulation to investigate the dynamic effect of binding on equilibrium constants, on the glass transition in proteins, and on ion pumping dynamics; and Jonathan Yewdell, NIAID/NIH, gave an overview of recent evidence that defective ribosomal products are the predominant source of class-I peptid ligands generated from viral and cellular proteins.

This three-day meeting was the first of a series on the next paradigms in computational and molecular sciences planned by Brown's new Center for Computational and Molecular Biology, whose aim is to establish a world-class center for research in the emerging field of computational biology; its central mission is to make breakthrough discoveries in the life sciences at the molecular and cellular level through the creative application of existing data-analytic methods, and the development of novel computational, mathematical, and statistical technologies required to exploit the opportunities emerging from advances in genomics and proteomics. The "next paradigms" meetings will feature speakers who are luminaries in their field talking about the grand challenges in that field, 'sweatshop' question-and-answer sessions, and innovative technical workshops. The Center's director and chair of the symposium, Sorin Istrail, believes that such meetings can serve as catalysts for leadership in research and education in computational and molecular biology.

The next meeting in the series will be held in October 2007; watch the website http://www.brown.edu/Research/CCMB/ for information on this and other upcoming events. Videos of the distinguished lectures as well as the follow-up sweatbox sessions of the December 2006 meeting is available at http://www.brown.edu/Research/CCMB/av.htm. **C!**

David E. Shaw

J. Craig Venter

Stephen Hoffman and President Ruth Simmons

Francofest
Excursions in Algorithmics:
A late Festschrift for Franco P. Preparata

Brown University, Providence, RI October 27-28, 2006

Organizers:
Nancy M. Amato
Der-Tsai Lee
Andrea Pietracaprina
Roberto Tamassia

A workshop code named Francofest and officially called "Excursions in Algorithmics: A Late Festschrift for Franco P. Preparata" was held on October 27–28, 2006, at Brown University to honor and celebrate Franco's outstanding career on the occasion of his 70th birthday (which took place in December 2005).

The presentations at the workshop covered an ample spectrum of research areas, a reflection of the breadth of the seminal results that Franco has achieved in over four decades of exceptionally productive research activity, shaping and giving momentum to multiple research fields.

Workshop attendees included current and formers students, colleagues, collaborators and friends of Franco's from all over the world and many mem-

> "The presentations at the workshop covered an ample spectrum of research areas, a reflection of the breadth of the seminal results that Franco has achieved in over four decades of exceptionally productive research."

bers of the Brown community. In a collegial and festive atmosphere, participants engaged in deep technical discussions on a wide variety of research topics, reminiscenced and shared anecdotes related to the times spent working with Franco and enjoying his company. At the end of the banquet, Franco addressed the audience with a subtly nostalgic talk entitled "Reminiscences of a Long Career" that zoomed back over his professional experience punctuated by memorable interactions with his research collaborators and friends.

The workshop organizers were former Franco students Nancy M. Amato, Der-Tsai Lee, Andrea Pietracaprina, and Roberto Tamassia. Ph.D. students Charalampos Papamanthou, Danfeng Yao, and Glencora Borradaile and administrative assistants Fran Palazzo and Janet Eager devoted a substantial amount of time to help with the organization of the event. Financial support for the workshop was provided by the Department of Computer Science and the Center for Geometric Computing at Brown University and by IAM Technologies, Inc.

For more information on Francofest, see http://francofest.net.

Franco at the University of Illinois

About Franco

Franco P. Preparata has been the An Wang Professor of Computer Science at Brown University since January 1991. Formerly, he was a Professor of Electrical Engineering and Computer Science at the University of Illinois at Urbana-Champaign. He received his Dr. Ing. degree from the University of Rome in 1959 and in 1969 was awarded the Libera Docenza by the Italian University System. After years of industrial experience with Sperry Rand Univac and Selenia, a subsidiary of Raytheon, he joined the faculty of the University of Illinois in 1965. Since then, he has also been a visiting professor at the University of Texas-Austin; the UFRJ, Rio de Janeiro, Brasil; the University of Pisa, Italy; INRIA, Rocquencourt, France; the University of Saarbruecken, Germany; Ecole Normale Supérieure, Paris, France; Kyoto University, Japan; Academia Sinica, Taiwan; the University of Padova, Italy; and the National University of Singapore, Singapore.

Franco has published more than 200 papers and is the author of three textbooks: *Introduction to Discrete Structures* (with R. T. Yeh), *Introduction to*

Logograms meaning "old very rare" and pronounced "Ko Ki" in Japanese. Calligraphy by Mrs. Yoko Muroga for Franco's 70th Birthday.

Franco with former and current Ph.D. students at Francofest. From left to right: Majid Sarrafzadeh, Gianfranco Bilardi, Charalampos Papamanthou, Roberto Tamassia, Franco Preparata, Andrea Pietracaprina, Vasiliki Chatzi, Nancy Amato, Ioannis Tollis, D. T. Lee

Franco at Brown University

Computer Engineering, and *Computational Geometry* (with M. I. Shamos). He is on the editorial board of six of the premier journals in theoretical computer science. Franco is a Fellow of the IEEE and of the ACM, and is listed in a large number of standard professional references. In 1993 he received the Darlington Prize of the IEEE Circuits and Systems Society. In 1994 he was a Fellow of the Japan Society for the Advancement of Science. In 1997 he received the "Laurea honoris causa" in Information Engineering from the University of Padova, Italy. Following early research in switching and coding, culminating in the discovery of the nonlinear Preparata codes, for the past three decades the focus of Franco's research has been the design and analysis of algorithms in their most general connotation. With the remarkable evolution of computer technology, his research interests have correspondingly evolved. He has been deeply interested in fundamental algorithms and data structures, VLSI computation and layout, and parallel algorithms.

Perhaps Franco's most enduring interest has been computational geometry, a spin-off of algorithmic research aimed at the systematic investigation of methods for the most efficient solution of geometric problems. Geometric problems are ubiquitous in human activities. Sporadic, and frequently inefficient, computer solutions had been proposed before, but in the mid-seventies computational geometry emerged as a selfstanding discipline targeted at this important area.

The goal of computational geometry is to analyze the combinatorial structure of specific problems as the underpinning of efficient algorithms for their solution. The field burgeoned, and in the mid-eighties Franco wrote a textbook on the subject with M. I. Shamos that helped establish it in the instructional arena. Today, an enormous body of geometric algorithms is known and this knowledge is increasingly indispensable in several applied areas such as geographic information systems, computer

graphics, and computer-aided design and manufacturing. Within the last area, Franco has also contributed to computational metrology—the assessment of the geometric quality of manufactured parts.

Today, Franco's main research focus is computational biology (also called bioalgorithmics), an emerging discipline that entails the development and use of mathematical and computer science techniques to solve problems in molecular biology—another example of computer science interacting with other fields. Since the discovery of the structure of DNA about fifty years ago and the digital underpinning of molecular biology, huge amounts of data have been generated in this field, making it necessary to resort to sophisticated computer science techniques for their analysis.

Franco lives in Providence, Rhode Island, with his wife, Rosa Maria. They have two daughters, Paola and Claudia, and three wonderful grandchildren. Franco's hobbies and interests include Zen gardening, stained glass making, history, archeology, and art. **C!**
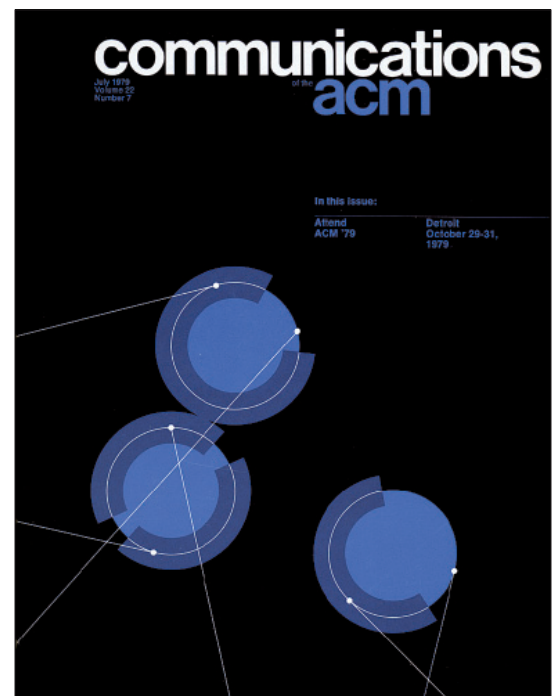
Cover of the July 1979 *Communications of the ACM,* inspired by Franco's paper "An Optimal Real-Time Algorithm for Planar Convex Hulls" in the same issue.

Parenthetically Speaking

First, I want to give a shout-out to fellow columnist Eugene Charniak. Eugene was recently made a University Professor, a rare and high honor that confirms that he is, indeed, a Mucky-Muck. Props, Eugene! (In my next column I will return to goading him.)

One of the arguable joys of being a program chair is dealing with unusual requests. I'm co-chairing CC 2007, a conference on Compiler Construction. CC resides within an umbrella called ETAPS, a confederation of conferences that meet every year in Europe. The advantage to working within the ETAPS framework is that ETAPS deals with the administrative details, from venues to publishing, leaving conference chairs to think thoughts of pure intellect.

Sort of.

I was recently contacted by a researcher who is a good friend who works for one of our leading technology corporations. (I will not even hint at their name: once you learn more about their lawyers, you'll see why. A wayward marksman is far more dangerous than a precise one.) He asked for the "confidentiality policy" that we follow. He clarified that they "are filing a patent on parts of the paper," so the lawyers were asking for the policy.

CC doesn't have a policy, and neither does ETAPS as a whole. It would be rather tricky for CC to have a policy, actually. Recall that ETAPS contracts the publishing? Once a paper is accepted, the publisher takes control of the content, and that is done on terms settled by ETAPS. Therefore, any policy that CC did adopt would probably have very limited applicability, and worse, would have the potential to directly contradict the contract signed by ETAPS.

My friend pointed out that both ACM and IEEE did have such a policy and pointed me to the policy of ACM SIGSOFT. It makes fun reading from this perspective. Note all the things it says about confidentiality:

http://www.sigsoft.org/about/policies/pc-policy.htm

As of October 2006: First, it describes a meta-rule. Then it discusses the confidentiality of reviews (but not of papers). Then it discusses the confidentiality of discussion (but not of papers). Then it provides a delegation obligation, but still doesn't say anything about papers. Then it makes a statement about access control, but not about confidentiality. And finally, it specifically addresses papers. To wit: "Neither SIGSOFT nor ACM guarantee the confidentiality of the submitted manuscripts."

This is the confidentiality policy that so satisfied the company's lawyers.

It's also worth noting the duration we're talking about. CC submissions are due mid-October. The decisions return early December, and final papers are due early January. This means the lawyers were trying to protect a window of not even three months. Of course, when it comes to the law, even a day can matter, but that begs the question: if this patent mattered so much, why risk exposure at all?

Given the number of conferences we have these days, chairs are, I think, expected to be utterly solicitous, cloying sycophants who will do anything for a submission. I couldn't resist. I told my friend that (a) we offered precisely the same confidentiality guarantee as SIGSOFT does, so it ought to please his lawyers, and (b) there was an extraordinarily simple and obvious way of protecting the confidentiality of the paper's content....

I spent November 2006 visiting researchers at the University of Edinburgh. Edinburgh is one of the world's centers of what I call Volume B theory, i.e., material that appears in Volume B of the *Handbook of Theoretical Computer Science, to wit, Logic and Semantics*. The numerous theoreticians in my own department, jointly and severally, pretty much ignore the very existence of this half of theory. Indeed, there were more computational logicians in the hallway of my secondary office at Edinburgh than there are at all of Brown.

Of course, research aside, being in Edinburgh is a magnificent experience, and we had a right royal bean-feast there. (That last bit is one of two phrases I picked up on that trip.) It's interesting to be away from Providence; I haven't lived anywhere else since moving here in 2000. Providence is a very pleasant place and a nice city; but Edinburgh is aggressively a city, and I've missed the bustle where we live in the quiet parts of Providence. It was also an opportunity to explore Scotland a little, though there

Shriram, when he's not touring the world on sabbatical

"While Hume was as rigorous and opinionated a thinker as any, he was also a bon vivant in the very best sense."

was a constant tension between doing that and sampling what Edinburgh itself has to offer.

You can read more about my Scotland experiences on my blog (follow the link from my home page), and look at some photographs. Meanwhile, here are teasers from my reviews of several Scotland-related books (whose full reviews are on my book review page):

How the Scots Invented the Modern World, Arthur Herman

Ridiculous title, but it is a compelling history of modern Scotland, particularly of the Scottish Enlightenment. I first read this book in 2002 and re-read it while in Edinburgh. My memory did not let me down: the early parts of the book remain some of the most lively popular history I have read.

Crowded with Genius, James Buchan

Buchan's book explores that crucial moment when Edinburgh became the crucible for the formation of the modern temper. It is a startling story indeed: how a dingy, dirty, provincial backwater suddenly glowed incandescent and, in a period of less than half a century, hosted notables—David Hume, Adam Smith—who rethought social and scientific foundations. Buchan's work is a study of this place and time. It is a book of formidable scholarship (the endnotes themselves make for fascinating reading), yet told with passion.

Millionaire, Janet Gleeson

An enjoyable book about one of the early financial bubbles.

Revolving Culture, Angus Calder

Echoing Beatrice Webb—that "people in the labor movement could be divided between 'As' (anarchists) and 'Bs' (bureaucrats)"—Angus Calder wonders if historians too might not similarly be classified, between "preservers of continuity" and those who "idolise intransigents." In that case, Calder himself should be classified as a 'C': a contrarian. Identifying strongly with the community of non-English English writers (such as C.L.R. James) and also

with the working-class Lawrence, Calder is an articulate, wise, and trenchant observer of Scotland and its place in the world.

Modern Scottish Culture, Michael Gardiner

Gardiner is a feisty cultural commentator concerned with the identity of Scotland. At the heart of his analysis is the distinction between nation and state, and how this applies to the Scottish identity within Great Britain: how what he terms the notion of Unionist Nationalism has played out in recent years.

The Emperor's New Kilt, Jan-Andrew Henderson

Henderson's book is a collection of anecdotes from Scottish history, many of them quite commonplace but a few eccentric. The kitschy title and cover, however, hide a somewhat smart book.

The Forgotten Hume, Ernest Campbell Mossner

Mossner's classic work (Columbia, 1946) on David Hume is subtitled "Le Bon David", the affectionate appellation given him by the habitués of Parisian salons. For while Hume was as rigorous and opinionated a thinker as any, he was also a bon vivant in the very best sense. The conversational salon culture he tasted early in his days he carried with him to Edinburgh, where he helped germinate the Scottish Enlightenment.

Walking Through Scotland's History, Ian R. Mitchell

When you read a sentence like "The demand for footwear is a constant aspect of the Jacobite advance and retreat which has attracted little notice from historians," you know you're reading a quirky and distinctive history.

Buddha Da, Anne Donovan

Jimmy has an announcement. He's become Buddhist. Not an announcement, really, just a mention as he's on his way out the door. Sure, he's one for taking a lark to something or another for a week or two, but when time passes and he's visiting the Centre even more often, life at home starts to come apart and get woven back together in new ways. **C!**

Unplugged

Eugene Charniak

Well, six months have passed since my last *Conduit* column, and if anything funny or unusual has happened in the department it has managed to escape my notice. So this is a book review column.

David Horowitz is a conservative writer whose idea of fun is to place an ad in the *Brown Daily Herald* listing reasons why reparations for slavery are a bad idea. I agree that it is fun to watch what happens and apparently so did the BDH editors, as they ran the ad. The results were predictable.

Horowitz's latest writing is a book entitled *The Professors: The 100 Most Dangerous Academics in America*. When I saw the book, I immediately looked to see if I made the list, but as I am not a left-wing nut-case I was not included. The book consists of short articles on one hundred professors. I cannot claim that I read every one—they get repetitious after a while—but I probably looked at more than half. There are no scientists at all, nor any Brown professors. One ex-Brown assistant professor made it; I presume he did not get tenure. So much for our reputation as a haven for left-wing tree-huggers.

"David Horowitz is a conservative writer whose idea of fun is to place an ad in the *Brown Daily Herald* listing reasons why reparations for slavery are a bad idea."

There is much silliness in the book, starting with the title. These folks are not dangerous; a lot of them are fruitcakes, and some are reasonable folk with ideas that Horowitz does not like. But there is an interesting and reasonable idea for a book here. It is too bad that Horowitz was not the person to write it.

The reasonable idea starts from the premise that academics are, as a group, left of center politically, just as generals, bankers and, I bet, members of all chambers of commerce are to the right of center. Conservatives make a big idea of the former fact, claiming overt discrimination, but the evidence for this is highly anecdotal and not very convincing. Nevertheless, the premise is true, and an interesting book would look at its causes and effects.

Actually, a serious listing of the one hundred politically nuttiest professors would be interesting, too. It is even possible that Horowitz has found them, but if nothing else the book's prose style is sufficient to give one doubts. Horowitz and his minions (he did not find or document all of the professors himself) never like to use a noun when a noun plus an inflammatory adjective would do. There are few professors, only radical professors. No politics, only extreme politics, etc. (As usual, I read the book a while ago, and I am depending on memory. That is why I did not use quotation marks in the last sentence.) In at least one case a professor is indicted for opposing the Iraq War. Maybe Horowitz should write a book on the forty million most dangerous voters in America.

I have my own theory to explain the preponderance of left-wing academics. Academics tend to like new ideas, and they are willing to give them hearings and try to evaluate them without undue bias. Conservatives tend to believe that we should be biased in favor of existing social institutions simply because they have stood the test of time. At this point my knowledge of political theory runs out, but already we should expect academics to tend leftward. Also, if I am right, there is no reason to believe that this should be a recent phenomenon. Probably the more egalitarian nature of academia today should imply comparatively fewer "upper-class" academics, and thus fewer with personal ties to the current social structure. Thus I would predict that the same bias existed in the past, but not to the same degree. I have no idea if this is true or not. If anyone knows of a study on this topic, let me know.

A second book I recently read is *The Ghost Map* by Steven Johnson. This one I strongly recommend to you all. It is the story of the two men in 19th-century England who are credited with the discovery and documentation of the cause of cholera. It has tragic victims, two great heroes, and interesting lessons on the nature of ideas and how facts are marshaled to support them. It does not have any villains, though, just misguided scientists who in retrospect held onto false beliefs way too long.

Cholera is not much in the news in modern life. It is a water-borne illness that does not affect anyone with a clean water supply. If that is not available, boiling works fine. But this was not known in the first half of the 19th century. People were certainly

> "Has tragic victims, two great heroes, and interesting lessons on the nature of ideas and how facts are marshaled to support them."

aware that environmental problems could cause illness, but the problems that affected large cities (which is where cholera outbreaks almost exclusively occurred) were generally thought to be airborne, not water-borne. As Johnson explains, in those times the most noticeable unpleasantness in cities was the air, and in particular the smell. Cholera bacteria are invisible and have no odor.

In 1854 an outbreak of cholera in London killed hundreds of people within an area of about one square mile. Two Londoners, Dr. John Snow, a physician, and Rev. Henry Whitehead, a clergyman, lived close to the affected area and took an interest in the matter. The book chronicles their efforts to discover the causal mechanism by which people became infected (by drinking infected water) and then to convince public-health officials first that they were correct and second to take action on the basis of their theory.

This story has many aspects worth highlighting. One interesting point is that the investigators were a team of a scientist and a clergyman. There have been a slew of books and articles recently on the compatibility or lack thereof of science and religion. Superficially, this team might support the side of compatibility, but it probably doesn't. What it does show is that Whitehead proved himself a very competent scientist, and that science is just clear thinking raised to the nth degree. It was Snow who proposed the waterborne theory and Whitehead who was on the local board set up to look into the outbreak. Whitehead was initially as skeptical as everyone else. (He had actually drunk the water himself and did not get sick. He had mixed it with whisky.) But Whitehead cared enough about the problem and was an exceedingly bright man. Eventually the evidence collected by Snow convinced him, and he thought up the critical "experiment" that clinched the case.

Snow had amassed data showing that one of the public water pumps had become contaminated. To a first approximation everyone who lived near the pump got cholera, and those closer to another pump did not. Whitehead extended this reasoning. Before the outbreak, nobody who used that water got sick, and after, everyone did. What happened in between? Whitehead's answer was that probably someone brought the illness to the neighborhood and contaminated the water. Who? When they phrased the question this way, Snow and Whitehead were able to identify the earliest deaths, and one of them had lived a few feet from the contaminated public water pump. From this they were able to find how waste from that house contaminated the water.

The "Ghost Map" of the title refers to a drawing created by Dr. Snow after the fact in order to convince the authorities: a map of the neighborhood showing the homes of everyone who died and the locations of all the public water pumps. It was also an early use of Voronoi diagrams, in that it drew boundaries at every point where two water pumps where equidistant, or perhaps we should say "equitime." (This is the closest the book comes to being relevant to computer science or mathematics.) Today the map is considered a striking piece of evidence; Tufte, the visual thinking guy, loves it. But as a piece of propaganda it did not work at all. While the local committee, on which Rev. Whitehead sat, came out in favor of the theory, the "expert" committee set up by the City of London was completely blinded by their preconception and still claimed that the disease was airborne. It took public health officials another ten years, and at least one more major outbreak, before the truth was generally accepted. At that point, the author notes, the experts rewrote history to show that everyone had known the truth ever since Snow and Whitehead did their pioneering work. Human nature never changes. **C!**

C! Comments?

Send your views to: Conduit,
Department of Computer Science,
Brown University, Box 1910,
Providence, RI 02912
or e-mail conduit@cs.brown.edu

Ping!

Where are you and what are you doing?

Let us know what's happening in your life! New job? Received an award?
Recently engaged or married? Use this form to submit your news or e-mail conduit@cs.brown.edu.

First Name

Last Name

My news:

Class Year

Address

City

State

Zip

E-Mail

Mail to: Conduit, Department of Computer Science, Brown University, Box 1910, Providence, RI 02912

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

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