



TRANSFORMATION

Computer Sciences

at Wisconsin



Table of Contents

Message from the Chair	2
Building a Top Department	4
Computer Sciences Research	6
Profiles	8
WES Program	26
Selected Alumni	28
Faculty	30



Tapping the Transformational Power of Computer Sciences at UW–Madison



Gurindar Sohi, Department Chair

One of the most satisfying aspects of the computer science field is witnessing how readily experimental advances translate into real solutions. Ideas find a home in the marketplace with remarkable speed and vigor. The competitive appetite for technological innova-

tion is enormous across society, whether it be in health and medicine, the biological sciences, commerce and industry, communication or government.

But the relationship between computer science and the afore-

At UW–Madison, we have learned to place a great deal of value in taking our research full circle, with a strong basic science focus followed by a commitment to applying and validating those ideas in real scientific and commercial environments.

mentioned fields has broadened of late. Rather than simply creating new products that improve processes, we are providing knowledge and techniques to fundamentally change those fields.

Rather than simply answering questions or solving problems faster or more accurately, we are opening these fields to new questions they have never fathomed before.

It's something we refer to at the University of Wisconsin–Madison as transformational technology.

Throughout our 43-year history, computer science at UW–Madison has been contributing to this culture of transformation. The department grew directly out of the mathematics department in 1963, and began to build upon its

theoretical foundations as new opportunities arose. It established itself as a national leader through the 1980s and 1990s in nascent fields such as database systems, computer architecture and distributed computing.

At UW–Madison, we have learned to place a great deal of value in taking our research full circle, with a strong basic science focus followed by a commitment to applying and validating those ideas in real scientific and commercial environments.

Let me offer a few examples of transformational thinking in the field:

Emeritus Professor Carl de Boor, our National Medal of Science winner, and member of both the National Academy of Engineering and the National Academy of Science, had an enormous impact with his study of splines, the mathematical constructs used to represent curves and surfaces in the computing realm. De Boor's simpler approaches to complex spline calculations underlie much of the present-day work in computer graphics and airplane and automotive design.

Professor David DeWitt, a member of the National Academy of Engineering, pioneered the development of database system techniques in the 1980s that are the underpinnings of most modern computer databases. DeWitt's group was among the first to develop highly scalable parallel database systems using commodity

processors, rather than exotic hardware, thus greatly improving the ability to store, manage and search data in both high-end and everyday computer uses.

Professor Miron Livny has taken distributed computing to new heights with his development of Condor, a novel grid computing system that offers supercomputing power with commodity hardware. This project now manages the computations of hundreds of scientific and industry user groups, harnessing the power of tens of thousands of desktop as well as rack mounted computers to run high-end computations in genomics, physics, health care, insurance and entertainment.

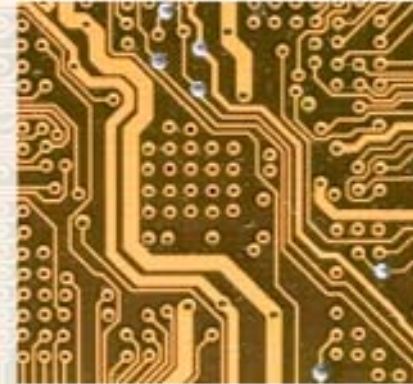
Emeritus Professor Olvi Mangasarian was among the first to apply optimization and data mining techniques to medical applications — in this case, developing better methods of breast cancer diagnosis and prognosis. His work has led to far less invasive alternatives to biopsy that yielded a diagnosis with 97 percent accuracy, and also classified the cases that will benefit most from chemotherapy.

Adding to the sense of current optimism in computer science is the recognition that this is a young, still-emerging discipline. We have only scratched the surface of our potential as a transformational science. Biology and medicine, especially, will present the next great opportunities. In the next decade, we can expect to see major computational forays into digital medical labs, personalized drugs and treatment plans, and the simulation of complex biological processes — all of which will benefit our quality of life.

This publication will provide, in greater detail, a look into current computer science research at UW–Madison across a broad spectrum of research areas. Many of these projects are taking on a transdisciplinary flavor on a major research campus like UW–Madison, with collaborations across a wide spectrum of disciplines. As one of only three institutions in the nation with top-ten programs in both computer science and the biological sciences — plus a major research-oriented medical school on the same campus — UW–Madison is poised to play a leading role in that next great transformation.

MEASURES OF EXCELLENCE

- Ranked the 10th best doctoral program in the United States in 1995 by the National Research Council (NRC), the most respected organization reviewing the nation's doctorate programs.
- Ranked the ninth best computer science program in the nation in the 2007 *U.S. News and World Report* ranking of America's best graduate programs, up one point from its last ranking of 10th in 2002.
- Recognized nationally as having the world's leading research specialty groups in computer architecture, database systems, distributed and grid computing, and nonlinear optimization.
- Two faculty — Carl de Boer and Stephen Kleene — have received a National Medal of Science, one of the most prestigious awards given in the sciences.
- The department is fifth in the nation in the total number of doctoral graduates placed on the faculty of top-25 computer science departments. The department placed 20 such graduates in the past decade. Only MIT, UC–Berkeley, Stanford and Carnegie Mellon produced more.
- The department has produced 17 winners of the National Science Foundation's most prestigious grant awards, including the Presidential Young Investigator, the NSF Young Investigator and the Career awards.
- Eleven faculty are Fellows of the Association for Computing Machinery (ACM) and four are Fellows of the IEEE.
- Four faculty have won the ACM Doctoral Dissertation Award. Only MIT, a much larger department, has as many winners of this award.
- More than 3,500 CPUs fuel the UW–Madison Condor project, a 20-year-old initiative that is now a world leader in high-throughput computing and has attracted hundreds of sites and thousands of users in both academia and industry.



Computer Sciences at UW–Madison: Building a Top Research Department



Initially started in 1963 as the Department of Numerical Analysis, UW–Madison became home to one of the oldest computer science programs in the country. And much like the field itself, the department has encountered some unexpected and remarkable growth spurts over the last four decades.

The early strengths of the department were in numerical analysis and applied mathematics, the foundation of what is now known as computational science. The department enjoyed strong national recognition with highly prominent faculty members, such as National Medal of Science winner Carl de Boer, whose work on simplifying spline calculations paved the way for their ubiquitous use in computing; and mathematical programming pioneer Olvi Mangasarian, whose famous “Mangasarian-Fromovitz” constraint qualification led to wide-scale usage in optimization theory.

Around the mid–1970s, some new areas of computer science were just starting to emerge. Databases were unknown at that time. Networking was only beginning as a field. That marked a critical era in the department’s growth, as leadership took the opportunity to test the waters, hiring young faculty teams around these emerging fields.

A concerted hiring effort during that period yielded a number of faculty who would go on to become defining forces in these now-entrenched fields. One example: Hired in 1976, David DeWitt was one of the early pioneers of database development, which soon led to a multi-faculty research team at UW–Madison that would become widely regarded as the top database group in the world. Major corporations such as Microsoft have benefited from the employment of scores of UW–Madison-trained database experts. And during this period, the department, with leadership by Larry Landweber, gained international prominence in networking, playing a major role in the development of the global Internet.

By the early 1980s, a second infusion of new faculty talent was triggered by the National Science Foundation’s “Coordinated Experimental Research (CER)” program. That program marked an unprecedented investment in select computer science programs across the country, in response to competitive challenges in Japan and other parts of the world. UW–Madison was one of the first five programs in the nation to receive CER funding — money which literally helped shape the face of modern computer science research. At a time when the cost of just a single computer was several times the salary of a full professor, this federal infusion of both new talent and technology from CER helped transform the department.

The hiring surges helped establish UW–Madison’s leadership not only in database research, but in computer architecture, another field where UW–Madison is frequently cited as the best in the world. Beginning in the early 1980s, CER funding helped launch a multi-year project called Crystal, one of the nation’s first forays into

parallel and distributed computing. This project became the forerunner of today's practical grid computers.

By the nineties, the young faculty from those hiring surges in the previous two decades really started to come into their own. The work of collaborative scholarship on the leading edge of new disciplines matured into a very productive era of discoveries and applications. In 1995, the National Research Council cited UW–Madison as a top-ten player in computer science.

In the 10 years since that designation, the department has made quantum leaps in many of its areas of excellence, and espe-

cially in the area of distributed computing, with the Condor system internationally recognized as a leader in research and

development of grid technologies. Medical and biological science collaborations have also blossomed, with projects helping improve cancer treatments, determine the structure of proteins and map

complex genomes. Data mining, machine learning and computer security have also become pillars of research strength over that time.

The evolution of a top-ten computer sciences program here is primarily the result of hiring strong faculty and retaining them at very high levels. UW–Madison has benefited from a “grow your

own” culture in the department, where young assistant professors receive mentoring and support on the tenure track, translating to higher tenure success and longer retention. Many colleagues across the country are surprised to hear that, out of the 60-plus computer sciences faculty hired since 1975, only two were hired with tenure.

While Midwest universities may not evoke the stereotypic image of an East or West Coast computer science hub, UW–Madison built its own recognition — with outstanding faculty, strong leadership and a knack for good timing.

The evolution of a top-ten computer sciences program here is primarily the result of hiring strong faculty and retaining them at very high levels.

FOUNDATIONAL FIGURES: SELECTED FORMER FACULTY

Carl de Boor

Theory and application of spline functions

Michael Carey

Database system performance

George Collins

Computer algebra systems

Anne Condon

DNA computing and randomized algorithms

James Goodman

Snooping cache coherence protocols

Yannis Ioannidis

Database query optimization

Stephen Kleene

Mathematical logic and recursion theory

Lawrence Landweber

Networking and the CSNET project

James Larus

Languages and compilers for parallel computers

Udi Manber

Applied algorithms

Olvi Mangasarian

Nonlinear programming

Edward Moore

Automata theory and Moore machines

Seymour Parter

Numerical solution of partial differential equations

Stephen Robinson

Nonlinear and stochastic optimization methods

Barkley Rosser

Lambda calculus

Leonard Uhr

Parallel architectures for AI

Computer Sciences Research



Artificial Intelligence and Computational Biology

Machine learning research focuses on both logical and statistical approaches, including “advice taking” and semi-supervised methods. Application areas include statistical natural language processing and computational biology. Collaborations with faculty members in the biological and medical sciences include genetic-microarray analysis and design, protein structure determination, and information extraction and privacy protection in medical text records.

Computer Architecture

This group has made myriad contributions to computer architecture including advances in branch prediction, caches, coherence, decoupled superscalar architectures, memory consistency models, out-of-order execution, simulation, pipelines, speculative multithreading, workload characterization, and many other areas that have impacted commercial designs and products.

Computer Graphics and Vision

Graphics work includes a broad range of topics: human figure ani-

mation, automatic video editing, crowd and group simulation, stylized animation and rendering, real-time and photorealistic illumination, and visualization of biological data. Themes in computer vision include medical image analysis, face recognition, and learning for vision.

Database Systems

One of the oldest and most accomplished database programs in the nation, research topics include query optimization, Web database systems and information integration, deductive databases, data mining, parallel database systems, scientific databases, visualization of large datasets, and database performance evaluation benchmarks.

Networks and Security

Research on computer networking includes wireless, mobile and distributed systems, security, network protocols, management approaches, and performance analysis. The Wisconsin Advanced Internet Laboratory (WAIL), is a one-of-a-kind facility for conducting network and distributed systems research. Software security research includes use of logic-based formal methods for tackling problems that arise from vulnerabilities in software.

Operating Systems and Distributed Systems

Operating system and distributed systems research includes the Condor high-throughput computing project; a “fuzz testing” technique for feeding random input into applications; building gray-box operating and storage systems; dynamic kernel instrumentation; and the Paradyn project, which has developed techniques for program instrumentation and automated performance analysis that aid the development of high-performance, scalable, parallel and distributed software.

Optimization and Numerical Analysis

The optimization group’s research includes the theory of optimization; generalized equations for describing equilibria and complementarity; algorithms and theory for nonlinear and combinatorial optimization problems; and use of distributed computing to solve very large computational optimization problems. Application areas include data mining and machine learning, radiation treatment planning for cancer, economic policy, and engineering design. Work in numerical analysis focuses on rep-

resentation and approximation of scientific data, including spline approximations, scattered data approximations, polynomial interpolation, and wavelet and Gabor representations.

Performance Modeling

Research on modeling and analysis for computer systems and communication networks involves the development of analytic techniques and their applications in system design.

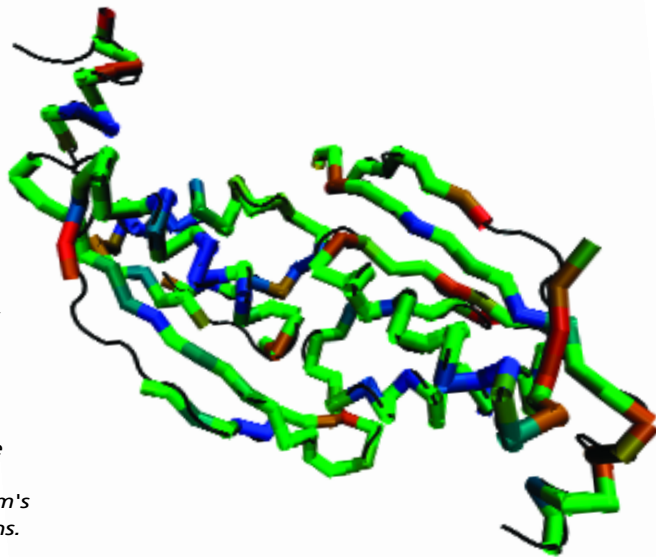
Programming Languages and Software Engineering

Research efforts include the static analysis of programs using 3-valued logic, program dependence graphs,

interprocedural dataflow analysis, and program slicing; the Wisconsin Safety Analyzer, which develops analysis techniques especially suited for commercial off-the-shelf components; compiler design using assist threads; and the Cooperative Bug Isolation Project, which helps track post-deployment software bugs with real users.

Theory of Computing

Current areas of interest include: algorithms, especially in number theory, algebra, and biology; computational complexity, especially structural complexity and the role of randomness in computation; emerging paradigms, especially quantum computing; and cryptography.



Shavlik's group is developing algorithms for locating atoms in 3D images of proteins. The thin black ribbon shows a protein's true "backbone," while the thicker colored lines indicate their algorithm's probabilistic predictions.

Research in Progress: Profiles of faculty vision, innovation

Could a new generation of computers become capable of taking advice, rather than commands?

Can new database systems empower scientists with unprecedented control over data from the human genome?

Could optimization techniques lead to a new world of secure patient record-sharing across the massive health care industry?

What will computing chips look like in 10 to 15 years?

These are among the many questions being explored today in UW–Madison's diverse computer science research portfolio. And they all feed into a much bigger question: What will the next big computer science transformation look like?

Wisconsin brings to the race a combination of depth and breadth in its research efforts. The CS department's 36 faculty conduct work in well-established disciplines such as computational theory and optimization, while also breaking ground in new fields such as multiprocessor architecture, security and computational biology.

The following pages will look at the intellectual life's work of nine individual faculty. Each profile will attempt to capture the pursuit of new ideas, new challenges and new scales that are driving continual change in the computer science field.

The panoramic view of UW–Madison CS research can be found throughout this publication. The profiles provide close-up snapshots in time.



“People were using really exotic technologies — CCD memories, bubble memories, all sorts of custom hardware. All this focus on custom technology meant we were always behind where the general computing field was going.”

Bringing database technology's ideas to grid computing

UW–Madison's leadership in the database development field started with a bit of pure luck in the mid–1970s, when new computer sciences professor David DeWitt was asked to teach an undergraduate course on the topic.

Having had only one graduate course on databases under his belt, DeWitt plunged into the available research on this new field. He didn't expect his interest in databases to go much beyond teaching a sound course on the topic.

But DeWitt began to take notice of a growing number of research papers looking at database systems and parallel computing, his true interests areas. That intersection of two studies triggered a research focus at UW–Madison that has become widely recognized as the nation's best in the database field.

The early research was high-impact because data is the ultimate currency of computing. Building these early systems also proved vexing.

“The field at the time was focused on hardware technologies, which was really the wrong focus,” DeWitt says. “People were using really exotic technologies — CCD memories, bubble memories, all sorts of custom hardware. All this focus on custom technology meant we were always behind where the general computing field was going.”

DeWitt's research group recognized that it needed to start using commodity processors and disks to be successful in creating highly scalable parallel database systems. To that end, the group launched the Gamma Project in 1984, and within three years were able to produce a truly scalable system that used mainstream computer technology. Many of the basic data flow and partitioning techniques from Gamma are alive and well in today's database technology.

“The key feature of Gamma was the use of hashing to divide up work tasks across a parallel environment,” DeWitt says. “It basically was a divide and conquer strategy: take this one big query involving data distributed across hundreds of processors, and execute the query in parallel without any centralized control.”

In the early 80s, UW–Madison also developed the field's first benchmark for evaluating database system performance, known simply as the Wisconsin Benchmark. That program was the precursor to today's widely used benchmarks such as TPC/C and TPC/D. The group then maintained a long-term commitment to studying performance evaluation in a way that has influenced how the entire database community conducts research.

DeWitt's research interests expanded in the 1990s into new areas such as database system performance and benchmarking, and distributed database systems. Other completed research projects in DeWitt's repertoire include Paradise, a parallel database system designed to support large geographic and image data sets; and Niagara, a distributed database system for querying distributed repositories of XML documents on the Internet.

Today, a great deal of DeWitt's research is homing in on what might be the field's greatest opportunity: Creating robust database systems for use by the scientific community. The field may even need to come up with entirely new models for dealing with the semi-structured or unstructured data sets of scientists in fields like bioinformatics and genomics. DeWitt has been working on that theoretical challenge in 2005–2006, while on sabbatical at the Massachusetts Institute of Technology (MIT).

One practical answer may lie with a current partnership between DeWitt and UW–Madison computer scientists Jeffrey Naughton and Miron Livny, to build a database system for the Condor grid computing community. Researchers using Condor rely on the distributed computing system to run massive computational challenges, but are generally on their own with data storage.

But building database technology into Condor could solve a major problem for the science community, DeWitt says. Scientific users have limited control over their data, which is the life's blood of their research. They manage data now by organizing it into elaborate sets of folders and directories, organized typically by type of data and date of collection.

“They end up with these enormously complicated collections that have semantic meaning, but are really sort of dead ends,” he says. “They can't run effective queries against that data.”

If they could harness the power of Condor to take over the entire computational and data aspect of big projects, it would provide tremendous value to the research community, he says. It could greatly improve workflow and scheduling and optimizing database queries. Much like the model followed with Gamma, DeWitt says the research is employing commodity software tools such as Web servers and application servers to build this system.

“Managing scientific data is one of the great opportunities moving forward,” says DeWitt. “We haven't done a good job so far for scientists. They have very, very large data sets and very, very primitive tools for managing them. We're really focused on solving that problem.”



DAVID DEWITT



JUDE SHAVLIK

“By giving computer systems more autonomy, they will be able to more quickly adapt to novel situations, they will be much more easily trainable by non-programmers, and human interaction with computers will become much more natural.”



‘Advice-Taking’ as a new curriculum for machine learning

As influential as machine learning techniques have become over the last two decades in managing big computational challenges, the learning styles employed remain relatively primitive: command, reinforce, correct, repeat.

Computer scientist Jude Shavlik is working on a new generation of machine learning that relies on a much richer and more natural set of training strategies, both for the computer and the user. Rather than the traditional input-output pairs, Shavlik envisions human teachers being able to give advice to a learning computer system.

Much like a human learning environment, the advice could be in everyday language and not precisely specified; and the computer system could be designed to accept the advice, reject it or even ask for additional advice.

In short, Shavlik wants to turn computers into savvier students.

“It is likely to prove highly valuable if we switch from commanding our computers to advising them,” says Shavlik. “By giving computer systems more autonomy, they will be able to more quickly adapt to novel situations, they will be much more easily trainable by non-programmers, and human interaction with computers will become much more natural.”

The advice-taking metaphor could help solve the challenge of creating software systems that are more adaptable to the unique user. Altering older systems to handle new tasks is very difficult, and everyday users who lack expertise can do little to personalize their software, Shavlik says. Human-to-machine communication is also limited in its current state, only offering a small range of possible inputs and outputs.

And who would the likely users be? Certainly the ideal test-bed for the approach would be in programs that employ simulation. “Tasks with video-game-like behavior have the sizable advantages of being visually oriented, so humans can more easily think up advice by observing behavior.”

Other examples include programs that deal with information finding and filtering, which can be trained by users to manage data; and programs that can evaluate and redesign scientific experiments on the fly, based on prior results.

The new paradigm is almost certain to find some applications within computational biology as well. Over the past decade, Shavlik has done an enormous amount of work with biostatistics and medical informatics, and is currently co-director of a National Library of Medicine training program in computational biology that trains 20 doctorates and post-doctorates each year.

Shavlik has a collaboration with UW–Madison biochemist George Phillips to help determine the three-dimensional structure of proteins, a critical question in determining how proteins bind and how they interact and “dock” with molecules. The function of proteins very closely follows structure, so the work has great implications in areas such as drug development, customized medicine and disease diagnosis.

The great challenge is in speeding up the process of defining protein structure. In Phillips’ lab, the team uses electron crystallography to create a 3-D electron map. That process may have once taken a multi-year doctoral project to accomplish, but now it takes months. Shavlik would like to compress the process to days.

Machine learning is capable of rising to the challenge. The group is training the computing program against known protein structures to help predict the structure of new ones. One goal is being able to use lower-resolution images from the crystallography process and still achieve accurate results. Shavlik says the statistical models are reaching a high level of accuracy in pinpointing protein structures.



“Our group has a computational bent in that we like to solve real problems... Many of our optimization peers are in math and engineering departments, but being part of computer science has really put us at the nexus of a lot of interesting interdisciplinary collaborations.”

Pushing the limits of optimization

From the molecular tangle of a folded protein to the sprawling national electrical grid, computer scientists at UW–Madison are applying optimization techniques across amazingly diverse scales, helping bring greater understanding to complex systems.

The department’s optimization faculty group is widely recognized as a world leader in optimization, which strives to minimize or maximize a function by manipulating a set of variables while respecting certain restrictions on those variables. Optimization techniques are used in a great variety of contexts, including finance and economics, weather prediction, engineering, biology and medicine.

“Our group has a computational bent in that we like to solve real problems,” says computer scientist Stephen Wright, a member of the UW–Madison optimization team. “We’ve been able to tie together the whole business of modeling the application and solving the mathematical problem. Many of our optimization peers are in math and engineering departments, but being part of computer science has really put us at the nexus of a lot of interesting interdisciplinary collaborations.”

One highly fruitful focus for the group, in partnership with medical physics experts at UW–Madison and nationally, has been in automating radiation treatment plans for cancer across a wide range of therapies. Optimization has brought greater precision and safety to brachytherapy or “seed radiation” techniques, as well as external-radiation techniques such as those delivered by the Gamma Knife or the Wisconsin-manufactured Tomotherapy treatment device.

“The optimization problem in radiation therapy consists of finding the ideal angles for the beams, and the ideal doses of radiation from each beam, to attack the tumor without hurting healthy tissue,” Wright says. “We set up mathematical models that describe the patient’s body and describe the effects of each beam on the patient’s tissues, then find the solution by applying optimization techniques to these models.”

In the Gamma Knife work, for example, UW–Madison computer scientist Michael Ferris was able to optimize a device used in hundreds of hospitals for treating brain tumors. The treatment is staggeringly complex, as the device uses more than 200 radiation beams to create a “spherical ball” of radiation inside the patient’s skull. Ferris’ work investigated how to place the balls of radiation so as to cover the tumor with a specified dose while avoiding damage to healthy brain tissue. Numerous doctors are now using treatment plans that are aided by Ferris’ optimization techniques and software.

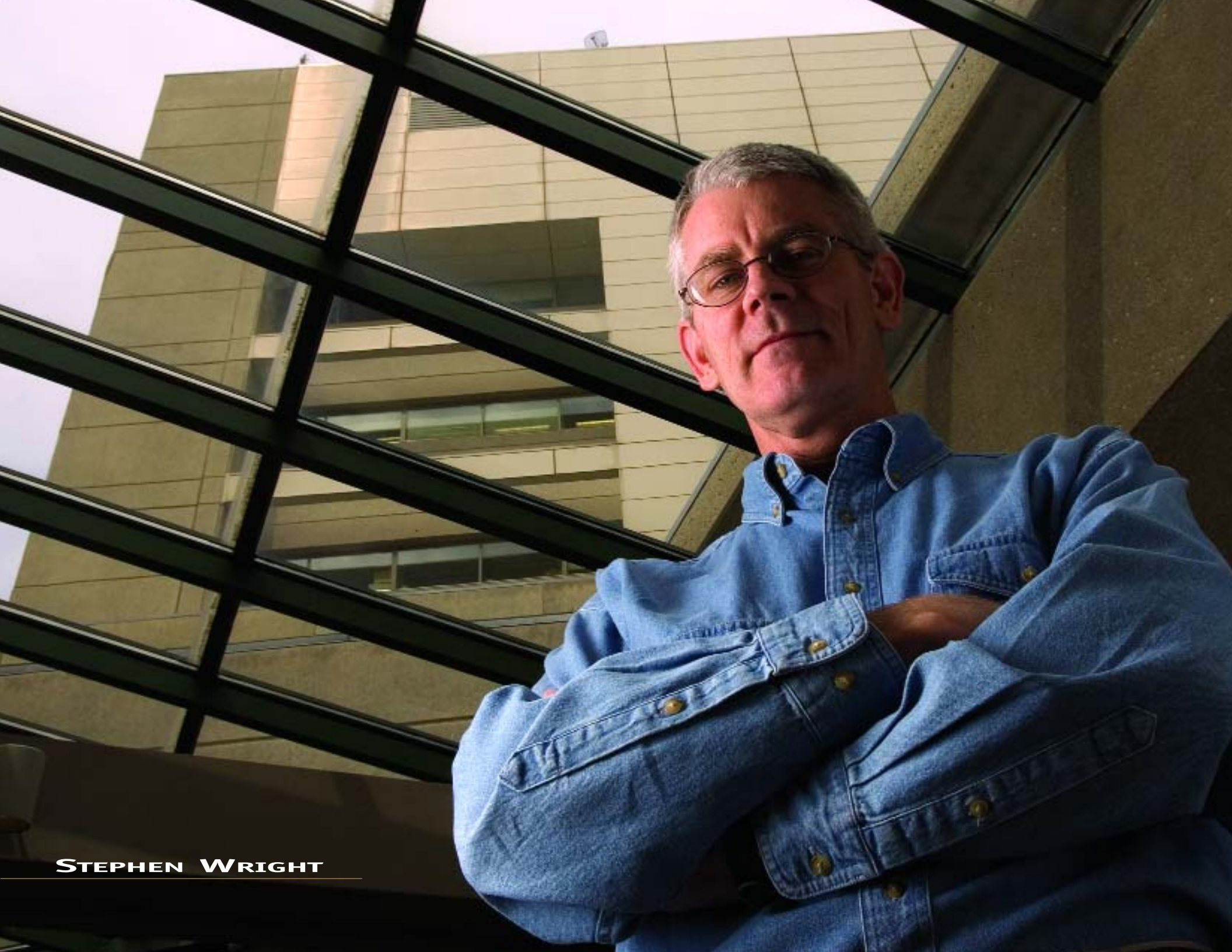
Wright has been working for more than a decade on another vexing challenge: bringing greater safety and efficiency to chemical plants. Keeping a petrochemical plant humming along efficiently means monitoring quantities that are constantly fluctuating, such as the flow of materials, temperatures, and reactions. Optimization techniques can be used to control this labyrinth of interrelated processes and steer the plant toward some desired mode of operation. “The challenge for the optimization and control process is to analyze the data in real time and set the inputs and controls accordingly,” Wright says.

Further evidence of the versatility of optimization is in structural biology and protein docking, one of the core questions of drug efficacy. “Many physical systems tend to go to the state of minimum energy,” Wright says. “So the reason that two proteins ‘dock’ is that the dock has lower energy than the separated state. The challenge is to figure out how and where they bind by finding the configuration that minimizes the energy.”

On the other end of the scale is work with engineers on optimizing the nation’s electrical power grid. The challenge is not only in simulating such a system, but also in controlling the grid in a decentralized fashion to avoid cascading failures that lead to blackouts. This is a high-priority project funded by the National Science Foundation to prepare the grid for challenges during peak-energy usage and avoid blackouts like those that occurred along the East Coast in summer 2003.

Wright is especially excited about a new project in partnership with systems engineering and nursing school faculty that will help health care and insurance entities share information more efficiently. “The state of the art in information sharing through the health care enterprise is surprisingly low,” says Wright. “In many doctor’s offices, people are still writing notes on the backs of cards. Their computer systems often are not sharing information with other providers, even in their own network.”

The first step of the process for Wright and collaborators is to develop a computational model that will help quantify the advantages of information sharing across all facets of health care. The model will help the federal government and private foundations to design incentives for information technology adoption that yield the biggest bang for the buck. “Health care is 18 percent of the nation’s economy, and we think there are some obvious inefficiencies that we can help address.”

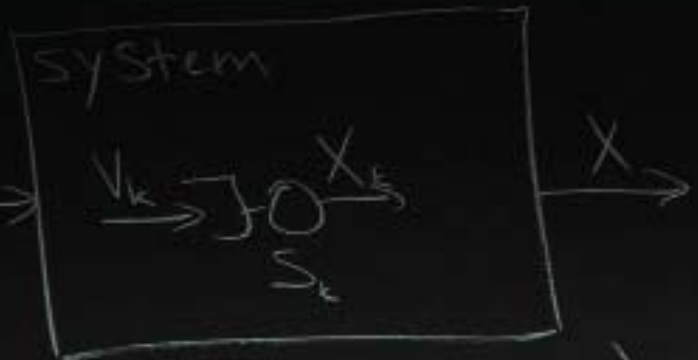


STEPHEN WRIGHT

Analysis

ck discover
y planning

sider h
sider l



(open or closed)

call: $X_k = V_k X$

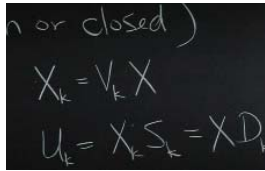
$$U_k = X_k S_k = X D_k$$

Consider

$$U_k \leq$$

\Rightarrow

“I advocate the use of these analytical equations because they have all the detail that matters, but not more detail than is needed... As compared with simulation, this approach is faster, and often more accurate, while still being adaptable to many questions.”



Handwritten mathematical equations on a chalkboard. The text above the equations reads "or closed)". The equations are:
$$X_k = V_k X$$
$$U_k = X_k S_k = X D_k$$

Computer systems design using analytical models

What if the path to faster, smarter, more powerful and more efficient computer systems could be boiled down to a simple, elegant equation — a mathematical recipe that tells designers how to get the greatest bang for the buck?

That premise is at the core of research by UW–Madison computer scientist Mary Vernon, who supplements conventional simulation techniques with highly abstract analytical models in the form of customized equations that point the way to peak performance. Her research is having a big impact in areas such as Internet performance and security, streaming media, computer memory systems and job scheduling on parallel systems.

The mathematical models are built by Vernon and her students “from scratch,” factoring in the mechanics of a system and its real-world behavior. Using versatile approaches, such as “mean value analysis,” they have consistently obtained new insights and new designs that are better than the real systems she is modeling.

“I advocate the use of these analytical equations because they have all the detail that matters, but not more detail than is needed,” Vernon says. “As compared with simulation, this approach is faster, and often more accurate, while still being adaptable to many questions.”

Vernon’s work has led to a great deal of application within the computer industry, including Intel, Cray and IBM. Some of her most exciting current work relates to overhauling the Internet Transport Control Protocol (TCP), the software that governs reliable downloading and is the backbone of all Web and electronic mail transactions today.

Without the TCP control mechanism, the Internet would be in greater danger of stalling and crashing, unable to manage the demands on bandwidth across millions of users. TCP helps manage the assignment of bandwidth to avoid congestion and collapse.

The current TCP, however, is more than a decade old and has a number of drawbacks that result in inefficient transfer and wasted bandwidth. The National Science Foundation (NSF) has placed a high priority on developing next-generation protocols for the Internet and has funded a number of competing university projects across the nation.

One of those is TCP Madison, and Vernon’s approach to the project could mark a sea change in how downloading is performed. The current system (named TCP-Reno) relies on transferring a window of information in 1,500-byte packets. The window gets larger as the transfer continues, which increases the speed of a download until a bottleneck occurs. Then

the window size decreases dramatically, to prevent congestion collapse.

Plotted on a graph, Vernon says the current download resembles a saw-tooth, scaling up, crashing down and repeating itself again. The system essentially causes information loss and then recovery, because it will ramp up downloading speed until a loss occurs.

Employing analytical models designed to correct this behavior, TCP Madison works to achieve convergence and stability in a download. The system will analytically determine available bandwidth share and quickly converge to the appropriate download rate, thus providing a steady and consistent transport of data with no packet loss.

Not only is it faster than competitors, Vernon says, but TCP Madison also is compatible with TCP Reno users, does not horde more than its share of bandwidth and requires no current change to Internet routers. TCP Madison is currently running on a large collection of hosts globally. “We’re getting incredibly good results,” Vernon says. “We cause no information loss and we go as fast as we can and we share fairly with others.”

Vernon recently completed another project trained on Internet security issues. Along with two undergraduate students, Vernon investigated the ways in which malicious online traffickers can sidestep the online monitors that are designed to identify and thwart potentially damaging attacks. A sophisticated user seeking to spread a virus via the Internet has the capability of mapping and avoiding those sensors, like a criminal figuring out what streets the police are assigned to.

Vernon’s research showed that such a map could be created in a very short time — a matter of days. The paper is helping industry recognize new ways to block malicious traffic, for example by hiding selective sensor data or moving them more frequently. The paper was named the best paper of the 2005 USENIX Security Symposium.

Vernon’s analytic modeling approaches also have led to some influential advances in the field of Internet video on demand. Those advances included near-optimal video streaming protocols that are widely recognized as the best protocols available today and produced three U.S. patents. The protocol optimizes the use of multi-casting, in which many users draw from a single video stream in order to reduce total demand on bandwidth. As copyright issues are better resolved, Vernon says video streaming on demand will become more viable for the mass market.



“It was a beautiful surprise to me ... to learn about Condor,” says Paolo Mazzanti, “We could save several months of work by simply using Condor. It is now our main way of using computer resources.”

Taking grid computing to new heights with Condor

After two decades of steady growth, Miron Livny's Condor “flock” has grown from a couple dozen networked computers to nearly 3,500 processors. But the most impressive “flock” in this powerful distributed computing network isn't really the computers. It's the users.

That was especially evident during the 2006 “Condor Week,” held in April at UW–Madison's executive conference center. The meeting attracted a venerable who's who of Big Science, including representatives from Fermi and Brookhaven national laboratories, the UC–Berkeley Space Science Laboratory and European high-energy physics powerhouse CERN. Also on hand were industry representatives from Fortune 100 companies like J.P. Morgan Financial and Hartford Life Insurance, and high-tech innovators like Micron and Yahoo!.

These representatives and more came to Madison to share how the Condor open-source software was accelerating their science or transforming their companies. The recent list even includes an animation company that used Condor to render millions of animated frames for the Disney film *The Wild*.

“Nothing surprises us,” Livny says of his growing legion of Condor users. He estimates Condor has about 1,600 deployments worldwide, managing at least 100,000 processors, churning out hundreds of years of computer time.

The Condor Project is devoted to harnessing the power of distributed computer resources to support high-throughput computing challenges — literally putting supercomputer capacity into the hands of mainstream users. Their unique software programs enable users to network computers and “scavenge” their available, untapped power to handle enormous computational tasks.

Livny has a unique take on the Condor research process. Likening the broad and diverse Condor usage to a massive clinical trial, Livny says his laboratory for distributed computing research is literally the environment where Condor is in use.

“The Disney movie and Hartford Life are basically our laboratory,” he says. “The usage is what's fundamentally important to what we are doing. Our team is unique in our approach and our commitment to the experimental side of computer science. This gives us an edge because we understand the side-effects of our ‘drug’ in a way no one else understands.”

The wide-scale adoption enjoyed by Condor has helped Livny maintain a 32-member research team and generate about \$4 million per year in federal research support. The National Science Foundation Middleware Initiative (NMI) has been a core supporter of Condor, recognizing it as “a mission critical tool” that not only efficiently solves current computational challenges,

but is capable of increasing the scale of future research.

“This (project) will help cause fundamental change in the way computing is used by scientists,” Livny says. “We are doing less science than we can because scientists are not aggressive enough in the way they use computing. Our project focuses on technologies that can empower research groups to pull together all their available resources.”

A prime example is the Grid Laboratory of Wisconsin (GLOW), a Condor-managed distributed system that employs more than 1,400 computers and delivers more than 30,000 hours a day of computing power. GLOW supports an amazing array of UW–Madison science, including optical genome mapping, astrophysics, cancer detection and treatment, and materials science.

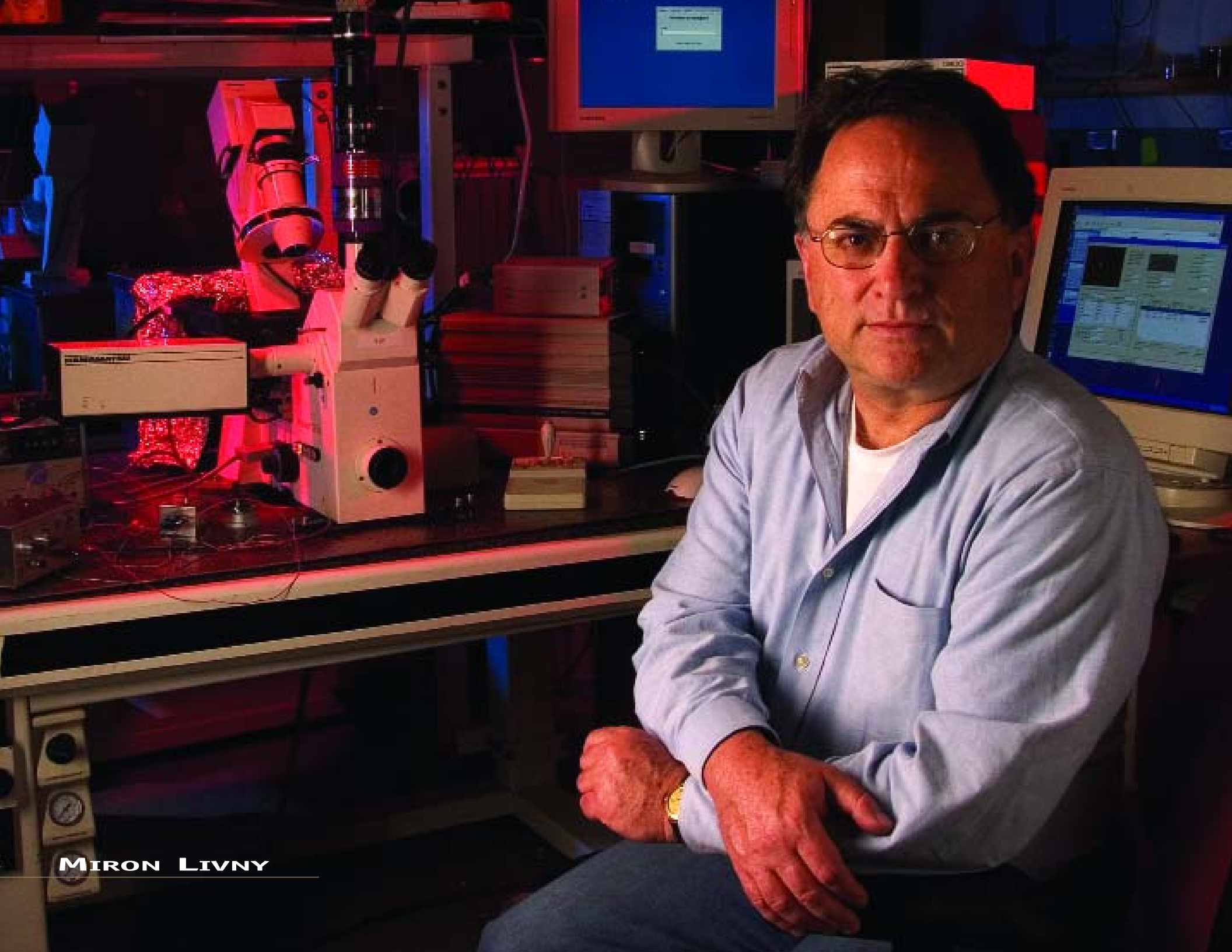
One of the most exciting Condor-supported projects helped demonstrate its versatility to scientists, Livny says. The 2003 project helped UW–Madison economics doctoral student Maria Ferreyra — now a professor at Carnegie Mellon — simulate the wide-scale effects of school vouchers on the nation's 20 largest cities. The project expanded on the relatively small number of existing programs to reach some broader conclusions about school voucher impacts. Ferreyra's powerful model used the equivalent of 50 years of computing time, Livny says.

The future looks very bright for Condor, due to the increasing complexity of science and an even greater imagination of the scale of distributed computing. Livny is currently working with a consortium funded by the National Science Foundation and the Department of Energy in creating a national Open Science Grid (OSG) that would follow the principles of UW–Madison's GLOW to offer high-throughput computing capabilities on a national scale. “We are positioned in a national and international arena to enable scientists to do things they haven't done before.”

That is one of the recurring themes of Condor users. With no supercomputers humming away in their basement buildings, many had not fathomed how they could put computers to work for them.

“It was a beautiful surprise to me ... to learn about Condor,” says Paolo Mazzanti, a physicist with the Italian National Institute for Nuclear Physics. “We could save several months of work by simply using Condor. It is now our main way of using computer resources.”

“It's fair to say that Condor has had a tremendous impact on the productivity of my research,” says another user, chemical engineer Fernando Escobedo. “The success and popularity of Condor could be quoted as a drawback for greedy users like myself, who for a couple of semesters got the dubious honor of ‘Condor Cycle King.’”



MIRON LIVNY



MARK HILL

“The best ways are when you are able to get people to work independently for long periods of time without a lot of information exchange. In the future, we will need that in our software.”



Harnessing Moore’s Law in a new generation of chips

Researchers in the field of computer architecture can be viewed, on one level, as the harnessers of Moore’s Law — the famous (and accurate) 1965 prediction that the number of transistors on a computer chip would double every 18 months. Given the constantly increasing resources available, computer architects insure that computing performance makes optimal use of all the added power.

For the past four decades, Moore’s law has played out through increasing transistors on a single chip applied to a single processor (or “core”), the mainstay of personal computing. But that’s about to change, and the biggest challenge for architecture researchers is going to focus on the new world where each chip contains multiple processors (called a “chip multiprocessor” or “multi-core” chip), and getting this new imperative for computer design in sync with Moore’s Law.

“The industry now realizes that throwing all of your resources into one processor will not continue to work,” says Mark Hill, who is part of UW–Madison’s renowned computer architecture research group. He notes that the recent Intel shift to the “dual-core” chip is a bellwether for major change across the personal computing landscape. “We’re at a real turning point. Programmers are facing a big challenge in making chip multiprocessors high-performance, and we don’t yet have a scalable strategy that allows for continued growth.”

That challenge has been embraced by UW–Madison’s computer architecture group, which already has more than a decade of research devoted to improving chip multiprocessor environments. Two major Madison projects — known as Multiscalar and Multifacet— are trained on designing hardware that will make this industry transition easier and more valuable.

Multiscalar, a project led by computer sciences chair Guri Sohi, started in 1995 and focuses on innovative ways to divide up work among processors to maximize their collective computational power. Dubbed “speculative multithreading” by the industry, Sohi’s techniques enable computers to distribute large chunks of conventional program code across many processors, without burdening programmers with authoring parallel programs. Sohi’s work involves having processors working more closely together than is common even today.

Multifacet, a project led by Hill and computer sciences professor David Wood, focuses on “using the transistor bounty provided by Moore’s Law to improve multiprocessor performance, cost and fault tolerance, while also making these systems easier to design and pro-

gram.” One exciting new offshoot of this project is called “Transactional Memory,” which allows programmers to specify “what” they want to appear atomic via “transactions” rather than specifying low-level details to coordinate with locks. Importantly, the higher-level transactional interface allows the implementation to seek the maximum concurrency allowed by the programmer’s intent.

The higher-level transactional interface also facilitates more implementation freedom. “By making transactions a thread-level abstraction, (transactional memory) permits implementations using different combinations of high-level software, low-level software and dedicated hardware,” Wood writes.

Computer architecture is an exciting research focus today, largely because any substantial leaps in computing performance will not come until the multi-core challenges are solved. Multi-cores also promise to make bug detection harder because program executions no longer follow a sequential path. “The hardest part will be dividing up the work in a way that’s balanced and doesn’t require undue exchange of information,” Hill says. “You don’t want the processors to be in a constant state of asking, ‘are you done yet, are you done yet’ ...”

Hill likens the architect’s challenge to the manager of an organization coordinating large numbers of employees. “The best ways are when you are able to get people to work independently for long periods of time without a lot of information exchange. In the future, we will need that in our software.”

The UW–Madison computer architecture group will be in a position of strength in the new research areas, given its track record of past theoretical and applied work. That includes advances in branch prediction, distributed shared memory, precise interrupts, pipeline clocking and many other areas. The group has also developed strong simulation tools, including SimpleScalar and Multifacet GEMS, which can test newly designed systems on how well they meet functionality, performance and cost goals.

One concrete measure of the strength of computer architecture at UW–Madison is in its doctoral graduates. A remarkable 15 architecture graduates since 1990 have gone on to take faculty positions at Top–25 CS programs, including Carnegie Mellon, Duke, Illinois, Maryland, Michigan, Penn, Purdue, Texas and Toronto. In addition, 14 of those 15 went on to receive NSF Career awards — and the one who hasn’t is a recent graduate.



“Honestly, anybody who has any chance of attacking these problems isn’t going to be motivated primarily by the money,” he says. “Besides, there are easier ways to make a million bucks.”

Truth, beauty and computational complexity

In the world of computer science research, there are questions and then there are Questions.

UW–Madison computer theorist Jin-Yi Cai is definitely intrigued by those capital-letter questions of his field. As a complexity theorist, Cai explores some of the fundamental laws of computation, where problems are big and overwhelming and may take an entire lifetime to solve — provided they can be solved at all.

While that may sound daunting, the thrill is in the chase for Cai and others in Theory of Computing (TOC). He describes his research with the language of an artist, driven by “elegance, internal logic and beauty.” The usefulness of the findings in this field can often be transformational, but they may not be evident until decades later.

“One of the great achievements in TOC is that people are able to find very unintuitive and novel methods for solving certain things,” says Cai. “And these ideas will generally not come to anyone who thinks about these things in a straightforward way. These evolve from a deeper understanding or inspiration. Sometimes you have the flash of an idea.”

Cai, a fellow of the Association for Computing Machinery, works in the area of computational complexity theory, a sub-area of TOC. The holy grail of TOC is “P versus NP.” The problem stands for “P” (easy to find) versus “NP” (easy to check), and it was named one of the seven “Millennium Problems” by the Clay Mathematics Institute (CMI). This organization has vowed to award \$1 million to anyone who can solve one of the seven intractable problems.

P versus NP is considered the greatest theoretical challenge of computer science. CMI describes it as “determining whether questions exist whose answer can be quickly checked, but which require an impossibly long time to solve by any direct procedure.” In other words, they may be problems that a billion advanced supercomputers will never have a prayer of solving in a billion years.

One such problem is the Factoring problem. Given a product of two large primes, how fast can one find its prime factors? This problem is believed to be one of those problems that cannot be solved by any fast algorithm (technically in deterministic polynomial time). However, people are nowhere near being able to prove such an assertion.

“Meanwhile,” Cai says, “researchers have found extremely clever methods which are much faster than any straightforward method — such as the Sieve of Eratosthenes — which divides the target number N by all primes up to \sqrt{N} . One should mention,” Cai continues, “that the problem of Factoring is different from the problem of testing if a number N is a

prime. The latter problem can be solved in time P — again by a non-trivial method.”

The “P versus NP” conjecture is that there are certain problems in NP that are not in P — in other words, they can never be solved by any polynomial time algorithm. But is it? That’s where the fascination lies for Cai. Some problems really may be proven to be lacking any fast algorithms, or there may be some strange, beautiful and ingenious method of cracking them. Cai doesn’t think the million bucks is the lure.

“Honestly, anybody who has any chance of attacking these problems isn’t going to be motivated primarily by the money,” he says. “Besides, there are easier ways to make a million bucks.”

Most complexity questions appear to exist on an almost philosophical plane. Yet they have proven “amazingly useful and practical,” Cai says. One area in particular is cryptography and security.

By finding problems that are so extraordinarily complex that it would take exponential time to solve them, theorists in this area essentially developed the backbone of modern computer security. “The whole notion that certain things should be difficult — for example, to break a vault — has an intrinsic value. In fact, all of electronic commerce today on the Internet requires security protocols which are based on the fact that some computational problems are difficult to do.”

While most people would think computer science originated as a discipline in the 1960s, Cai puts the date back to the 1930s, when scholars like England’s Alan Turing, Alonzo Church of Princeton and Stephen Kleene of UW–Madison laid the foundation of computability theory. Turing was also well-known for his work during World War II, helping break the codes of the Nazi Enigma machines that encrypted secret battle information. The “Turing test” is a simple and elegant way of measuring artificial intelligence. And the precise concept of algorithms — the backbone of computer science research — was developed by Turing and Church.

What sets computability apart from complexity is that the former is concerned with what is ultimately computable, while the latter is concerned with what is feasibly computable. “Our understanding of what is feasibly computable is still very primitive,” Cai says.

“In fact,” Cai says, “despite the ‘P versus NP’ conjecture, most progress has been made in finding ingenious and strange methods of doing certain computations fast, which you thought were impossible to do.”



JIN-YI CAI



REMZI AND ANDREA ARPACI-DUSSEAU

“There are companies that could lose millions if they took three days offline to restore information after a system failure...They need systems that keep running even when failures occur.”



Behind the façade of the black box

With so many enterprises whose very survival depends on their data, a new currency of modern businesses has become availability — as in 24/7 online availability, with zero tolerance for down time.

“There are companies that could lose millions if they took three days offline to restore information after a system failure,” says Remzi Arpacı-Dusseau, a UW–Madison computer scientist. “They need systems that keep running even when failures occur.”

And while your average desktop user may not have a business on the line, having data become corrupted or compromised is no less troubling.

Remzi Arpacı-Dusseau and fellow computer scientist (and spouse) Andrea Arpacı-Dusseau lead a team of about a dozen computer science graduate students at UW–Madison in exploring data storage reliability, security and other issues through the use of what they call “gray box techniques.” Essentially, it means peering into behaviors and patterns that occur between different computer components, rather than looking at “black box” systems in isolation. Often, it is at the interface of different systems where the biggest problems occur.

The research group collaborates directly with the multi-billion dollar information storage industry, including companies such as EMC, Network Appliance, Seagate and IBM, on helping make disk storage systems more reliable for consumers in both high-end and desktop environments. The group has shared ideas on developing semantically smarter disks and procedures that can help isolate failures.

These are core questions to this competitive industry. “People are starting to see more failures at a time of higher reliance,” Remzi Arpacı-Dusseau says. “We now have disk drives that are capable of holding 750 gigabytes — almost an entire terabyte. When I was growing up, we talked about megabytes being a lot of storage. There is tremendous industry pressure to keep piling on the bytes and still keep costs down.”

As a consequence, more failures are occurring and the current operating systems aren’t ready for the problems. Says Andrea Arpacı-Dusseau: “We’re trying to figure out how to make the next generation of operating systems more adept at handling the possibility of disk failures.”

One current project, called IRON (internal robustness) file systems, offers techniques that can improve the ability of operating systems to identify and circumvent common bugs that

occur in the disk interface. The program was developed by studying what happens when disk-related failures are inserted under operating systems. Unfortunately, what they have found are that disk problems often wreak havoc on the operating system itself, and prove to be one of the most common reasons for system crashes.

Another application from the Advanced Systems Laboratory (ADSL) team is called D-GRAID, a raid array of disk drives that degrades gracefully, rather than all at once. This addresses a major issue in networked databases. One common backup method today is called mirroring. If you have 100 disks in a networked database, people will take half of them and copy all the data to the other half. The problem is, the system can face catastrophic data loss if two or more undetected failures occur. D-GRAID solves the problem through careful placement of data, ensuring that even if some small number of disks fail, a large fraction of user data can be recovered.

Much of the work of ADSL is built around the concept of creating smarter storage and retrieval systems. Disk technology is generally the dumber member of the computer family, Remzi Arpacı-Dusseau says, and that’s mainly by design. The industry wanted to design disks that can be used universally, where their complex inner workings didn’t need to be understood by operating systems. “But the negative has been that it really prevents you from doing interesting things inside the disk drive. We’re trying to get back some of the things we lost from these dumbing effects.”

One of the value-added goals of the “semantically smart disks” research is in improving security in the technology. For example, when deleting a file, conventional disk systems do not have the capacity to truly remove that information. It is an enormous privacy issue with used, recycled or discarded computers, since important financial and personal information can be exposed. Remzi Arpacı-Dusseau notes a recent MIT project in which a researcher purchased 400 used disks off eBay, and proceeded to find all sorts of credit card numbers and personal identification.

Remzi Arpacı-Dusseau says that many of the nation’s biggest companies with huge database needs are learning how to grapple with all of these issues associated with disk storage. “The problem is,” he says, “the technology hasn’t percolated down to your desktop.”



“Software developers deploy their programs and rarely hear directly from users, but the poor guy in tech support gets an earful...That’s the only kind of feedback you get; you lob it over the wall and hope it works.”

Ganging up on computer bugs

Ben Liblit offers a bold prediction regarding all of the complicated software programs churning away in your computer:

They have bugs. All of them. Guaranteed.

“Software bugs are part of the mathematical fabric of the universe,” says Liblit, a UC–Berkeley doctorate who joined the UW–Madison computer sciences faculty in 2005. “It is impossible with a capital ‘I’ to detect or anticipate all bugs.”

The staggering complexity of software is only part of the issue, he explains. Software has so many different points of interaction — with hardware, with networks, with other software, and mostly with humans — that opportunities for buggy behavior abound.

“That behavior is so dynamic that it becomes useful to look at (software programs) almost like they were some sort of organic system, whose complete behavior is unknowable to you,” says Liblit. “But there are behavior trends you can observe in a statistically significant way.”

Liblit takes that metaphor literally in his research, known as the “Cooperative Bug Isolation Project.” Combining programming languages and software engineering with a dose of machine learning, Liblit has enlisted real software users to tell him where the bugs are. And he has a growing “kill board” as evidence the idea works.

Liblit has created lightweight instrumentation that is added into the binary language of software. The instrumentation creates a sparse but statistically fair random sample of software behavior, while also protecting user privacy. The system produces regular “feedback reports” across the thousands of software programs that are in use. All of those reports get fed into a powerful database that accumulates the data and starts to identify trends.

Then, through statistical modeling techniques, Liblit is able to pinpoint software bugs that are occurring with enough regularity to affect many users. The final step in the feedback loop is a bug report, prepared by Liblit and sent back to the software engineers capable of acting on the results.

The science behind this real-world debugging approach received national attention this year. The Association for Computing Machinery (ACM) named Liblit’s doctoral dissertation on cooperative bug isolation as the best in the world in 2005 among dissertations nominated from both computer science and engineering programs. And while the open-source software community has become an ardent adopter of the program, Liblit also has attracted interest from IBM and Microsoft.

The real excitement of the project, Liblit says, is that it could dramatically improve the ability to enhance software post-deployment. “Software developers deploy their programs and rarely hear directly from users, but the poor guy in tech support gets an earful,” he says. “That’s the only kind of feedback you get; you lob it over the wall and hope it works.”

This system provides direct information about real-world software problems that developers can act on with statistical certainty. Right now, the only way to gauge real-world performance is based on the “squeaky wheel” effect of those who file bug reports or call tech support. And real-world performance will always be an important variable in dynamic debugging techniques.

“It has been cynically observed that software developers use their consumers as beta testers,” he says. “I think there’s a lot of truth to that observation. The problem is consumers are not very good beta testers. They’re not very disciplined, they don’t keep good records, they never do the same thing twice. My solution is to make them better beta testers.”

Right now, Liblit has a number of users in the open-source software environment, where the application has been added to popular programs such as Evolution (similar to Microsoft Outlook), Gnumeric (a type of spreadsheet), Rhythmbox (similar to iTunes) and Spin, a CPU simulator in heavy university use. He has posted 192 versions of eight different open-source applications over the three-year life of the project.

The system averages just under 3,000 new reports per month, and bug rates vary a great deal across applications, he says, with “crash rates” (where the program shuts down) as high as 8 percent in one application to a low of 0.4 percent in another. All in all, his “kill board” has recorded 546 outright program crashes and 11,369 lower-level errors as of April 2006.

Liblit says user privacy is absolutely essential to the program’s future viability. The randomly sampled snippets of data have no identifiable information unto themselves, and only have meaning when aggregated in large numbers.

Will software development ever reach a level of sophistication that would render Liblit’s bug machine obsolete? Liblit says that scenario is unlikely. Given that software becomes vastly more complicated with each generation, his better guess is that the industry is maintaining an even keel.

“We’re trusting our software more than ever before,” Liblit notes. “But with the right technology, the users themselves can help make software better for everyone.”



BEN LIBLIT

Wisconsin Emerging Scholars: Working to build a diverse new generation of computer scientists

A two-year-old program at UW–Madison called Wisconsin Emerging Scholars in Computer Science (WES-CS) is exploring novel ways to spur interest in the field among women and minorities. But getting that longer-term commitment to degrees and careers in CS will continue to be a stubborn challenge.

Professor Susan Horwitz, associate chair of computer sciences at UW–Madison, created the WES-CS program with initial grant support from Microsoft, and now has National Science Foundation backing to continue. The program is combining two core strategies — direct recruitment of new freshman students from underrepresented groups, and parallel team learning techniques — to the department’s introductory-level course.

Horwitz says those two proven techniques had never been combined in a first-year CS course, and the strategy serves to both increase

the pipeline of underrepresented students and manage their quality of experience once enrolled.

Diversity is a serious issue facing the future of computer science. While science and engineering

Horwitz says the freshman-level infusion of practical computer science ideas will help students “gain a better understanding of the breadth of the field, the positive ways in which computer science can affect people’s lives, and the range of interesting career opportunities.”

fields across the board struggle to attract a true cross-section of students, the problem is most pronounced in computer sciences. At U.S. research universities, typically only about one in ten CS bachelor’s degree graduates are women, and the number drops closer to five percent at the doctoral level.

“The numbers are terrible for computer science and they have been trending downward so far this decade,” says Horwitz, noting that

UW–Madison women CS undergraduates have gone from 11 percent in 2000 to 9 percent in 2005.

“No one completely understands the trend. Some of it may stem from the dot.com bust and

a sense that outsourcing may be threatening future jobs. But we’re actually looking at a huge pending shortage in the computing workforce.”

In the first year of WES-CS, the department enrolled 48 total students in the program — 26 (54 percent) of whom were women, and six (12.5 percent) of whom were in underrepresented racial and ethnic minority groups. That compared to 22.7 percent and 8

percent, respectively, in the general population of the class. More than 50 percent of women in the WES program reported they would not have taken the course without direct recruitment.

For their efforts, the WES-CS students received an extra credit for a once-a-week, two-hour group session that focused on fun, interesting exercises well-suited to group problem-solving. This part of the class proved very successful, with 76.5 percent of respondents reporting the interaction deepened their understanding of the material.

However, those classroom positives did not translate to greater interest in a CS degree for many students. In the survey, 12 of 34 students said they were originally interested in a CS major but decided against it after the class. Of that group, two-thirds of them had the refrain: “I don’t want a job sitting in front of a computer all day.”

Those comments actually pave



WES-CS students play the roles of different variables in a piece of Java code to help them understand method calls and aliasing.

the way for opportunity, rather than discouragement, Horwitz says.

“The students were overwhelmingly positive about their involvement in WES, but they frequently followed that up with, ‘I’m majoring somewhere else.’ We hear

a lot that they want a career that is relevant. With women and minorities in particular, they are picking careers that will have impact and have ties to their communities.”

Horwitz and others note that computer careers are rich in rele-

vance and impact, but universities have not been actively conveying that message. To that end, Horwitz instituted a dinner and guest lecture program for WES-CS students, inviting in leaders from local computer companies to talk about the

profession. The program is going a step further in fall 2006, creating a campus “First-Year Interest Group” built around topics related to computers and real-world problem solving and service.

An innovative “Digital Divide” course will be added in fall 2006 that students take simultaneously with the introductory programming course. The class will examine the impacts of technology on different societal groups and will include a group community service project.

Horwitz says the freshman-level infusion of practical computer science ideas will help students “gain a better understanding of the breadth of the field, the positive ways in which computer science can affect people’s lives, and the range of interesting career opportunities.”

A great deal is at stake for the profession, she adds. All industries want a workforce that is representative of the larger society and incorporates a diversity of ideas and backgrounds. As computers become even more omnipresent in people’s lives, Horwitz says, diversity in the industry will be the only way to guarantee that all people benefit from the advances.

STEVEN SEITZ

STUDYING CS AT UW—MADISON



Steven Seitz is a 1997 doctoral alumnus of UW—Madison and professor of computer science at the University of Washington. His expertise is in computer vision and computer graphics. Steve took time recently to reflect on his graduate school experience at Madison.

What would you say were the key factors in choosing UW—Madison for your graduate study?

I was an undergraduate at the University of California, Berkeley, where my faculty advisors had a very high opinion of the University of Wisconsin, encouraging me to apply there for graduate school. I later called up faculty members at a bunch of schools and really hit it off with Chuck Dyer at UW—Madison. Shortly thereafter, I flew out for a visit and was really impressed with the faculty, the graduate students, and Madison as a city. That sealed the deal for me.

When did your focus begin to emerge in the fields of computer graphics and computer vision? Was it a topic of interest coming from Berkeley, or did it emerge at UW—Madison?

In my senior year at Berkeley, I took a computer graphics class and loved it. Ironically, the graduate applications had already been sent out, and most of the places to which I had applied were not well known for their graphics programs. Wisconsin in fact had no graphics faculty at the time (happily, this has now changed with the arrival of Mike Gleicher and the pending arrival of Li Zhang). After conversations with Chuck Dyer and his grad students, however, I became intrigued by the area of computer vision, and began working

with Chuck on vision problems in the summer of my first year at UW—Madison.

My fascination with computer graphics continued, however, and I ended up combining both of these interests in my graduate research. The timing was perfect, as research at the interface of vision and graphics started to blossom in the research community as a whole, and I was able to help make some of the early contributions.

Who were the faculty mentors at Madison who were critical to your research and eventually your dissertation? What was the topic of your dissertation?

Chuck Dyer was my advisor and mentor during my time at UW—Madison. He's a great collaborator and I thoroughly enjoyed our interactions. My thesis was on the topic of taking a collection of photographs and producing synthetic views of the same scene from new camera viewpoints.

From your perspective at Washington, what would you say today about the quality and reputation of computer science at UW—Madison?

UW—Madison computer sciences is an absolutely first-rate department in every possible way. I always strongly encourage my students to apply there. As a case in point, UW—Madison was the one place that I encouraged Li Zhang to apply this year for faculty positions, and now he will be part of UW—Madison's graphics team.

Anything else interesting, unique or cool about your time in Madison?

I shared a house with five other grad students down near the Vilas Zoo. Of those, four are now faculty members (at Carnegie Mellon, Washington, UT—Austin, and Waterloo), and one works at Intel. You'd think we would have had lots of interesting conversations about computer science, but I think we all spent most of our time in the basement with the Sony PlayStation.

Selected CS Alumni

From 1964 to 2005 the department granted 2,927 Bachelor's degrees, 2,434 Master's degrees, and 520 Ph.D. degrees.

1970s

Eric Harslem (M.S., 1968), former CTO, Dell
Judy Faulkner (M.S., 1970), CEO, Epic Systems
Alan Merten (Ph.D., 1970), President, George Mason University
Edward Robertson (Ph.D., 1970), Indiana University
Carol Bartz (B.S., 1971), Executive Chairman of the Board, Autodesk
Stephen Robinson (Ph.D., 1971), University of Wisconsin–Madison
David Wise (Ph.D., 1971), Indiana University
Tom Bruggere (M.S., 1972), former CEO, Mentor Graphics
Christoph Hoffmann (Ph.D., 1973), Purdue University
Tsong-Jen Huang (Ph.D., 1973), Managing Partner, AsiaVest
Rick LeFavre (Ph.D., 1974), former VP, Apple
Phillip Barrett (M.S., 1978), former CTO, Real Networks

1980s

Dina Bitton (Ph.D., 1981), VP, SAP
Russel Sandberg (M.S., 1981), Co-founder, Legato Systems
Thomas LeBlanc (Ph.D., 1982), Provost, University of Miami
Joseph Moran (M.S., 1982), Co-founder, Legato Systems
Pavan Nigam (M.S., 1982), Co-founder, Healthon/WebMD
Rakesh Agrawal (Ph.D., 1983), Technical Fellow, Microsoft Search Labs
Ellis E-Li Chang (M.S., 1983), VP, KLA-Tencor
Insup Lee (Ph.D., 1983), University of Pennsylvania
Vighneswara Mokkarala (M.S., 1983), Executive VP, Envestnet
Rob Hagens (M.S., 1984), former senior VP, Level 3
Jonathan Kepecs (Ph.D., 1984), Co-founder, Legato Systems
Matt Korn (M.S., 1984), Executive VP, AOL
Mark Shavlik (B.S., 1984), CEO, Shavlik Technologies
Jaiwei Han (Ph.D., 1985), University of Illinois
Pat Hanrahan (Ph.D., 1985), Stanford University
Peter Spiro (M.S., 1985), Distinguished Engineer, SQL Server, Microsoft
Brian Pinkerton (B.S., 1986), Creator, WebCrawler
Victor Shoup (Ph.D., 1989), New York University

1990s

Daniel Ralph (Ph.D., 1990), University of Cambridge
Todd Proebsting (Ph.D., 1992), Director, Center for Software Excellence, Microsoft
Steve Scott (Ph.D., 1992), CTO, Cray
Sarita Adve (Ph.D., 1993), University of Illinois
Vikram Adve (Ph.D., 1993), University of Illinois
Thomas Ball (Ph.D., 1993), Head, Software Reliability Group, Microsoft
Kristin Bennett (Ph.D., 1993), Rensselaer Polytechnic Institute
Manoj Franklin (Ph.D., 1993), University of Maryland
Michael Franklin (Ph.D., 1993), University of California, Berkeley
William Roth (M.S., 1993), VP, BEA Systems
Jeffrey Hollingsworth (Ph.D., 1994), University of Maryland
Kiriakos Kutulakos (Ph.D., 1994), University of Toronto
Renee Miller (Ph.D., 1994), University of Toronto
Kurt Brown (Ph.D., 1995), Director, Intel Research Berkeley
Joseph Hellerstein (Ph.D., 1995), University of California, Berkeley
Alvin Lebeck (Ph.D., 1995), Duke University
Vaishnavi (Anjur) Sashikanth (M.S., 1995), VP, Hyperion
Odysseas Tsatalos (Ph.D., 1995), CTO, oDesk
Todd Austin (Ph.D., 1996), University of Michigan
Mark Craven (Ph.D., 1996), University of Wisconsin–Madison
Robert Olsen (Ph.D., 1996), Distinguished Engineer, Cisco
Steven Reinhardt (Ph.D., 1996), University of Michigan
Steven Seitz (Ph.D., 1997), University of Washington
Douglas Burger (Ph.D., 1998), University of Texas at Austin
Babak Falsafi (Ph.D., 1998), Carnegie Mellon University
Andreas Moshovos (Ph.D., 1998), University of Toronto
Jignesh Patel (Ph.D., 1998), University of Michigan
T.N. Vijaykumar (Ph.D., 1998), Purdue University
John Watrous (Ph.D., 1998), University of Calgary
Johannes Gehrke (Ph.D., 1999), Cornell University

2000s

Anastassia Ailamaki (Ph.D., 2000), Carnegie Mellon University
Amir Roth (Ph.D., 2001), University of Pennsylvania
Jayavel Shanmugasundaram (Ph.D., 2001), Cornell University
Daniel Sorin (Ph.D., 2002), Duke University
Craig Zilles (Ph.D., 2002), University of Illinois
Milo Martin (Ph.D., 2003), University of Pennsylvania

Faculty Research Summaries

Artificial Intelligence & Computational Biology



Jude Shavlik
Statistical Machine Learning

Jude Shavlik focuses on creating computer algorithms that learn, both from their experiences and from instruction provided by humans. One major interest is the creation of algorithms that are able to accept, refine and extend a rich variety of instruction, ranging from labeled training examples to informal advice expressed in ordinary English. A second major interest of the Shavlik group is the application of machine learning to biomedical tasks such as protein-structure determination, preservation of privacy in medical databases, disease diagnosis, microarray ("gene chip") analysis and design, and information extraction from on-line biomedical text.



Xiaojin Zhu
Statistical Machine Learning

Xiaojin Zhu's research centers on novel approaches to semi-supervised learning that require less human effort and promise better results. Zhu devised

learning as data-dependent regularization. Zhu is interested in graphical models and kernel methods in general. On the application side, Zhu is working on statistical natural language processing, including sentiment analysis from consumer product reviews, and word sense disambiguation using WordNet and latent variable models. He is also investigating applications of machine learning to image processing, networks, program analysis, astronomy, bioinformatics, and psychology.

Computer Architecture



Mark Hill
Computer Architecture

Mark Hill's research in computer architecture targets the memory systems of both multiple- and single-processor computer systems. Memory system design is important because it largely determines a computer's sustained performance. In particular, Hill studies transactional memory, cache coherence, memory consistency models, cache design, parallel computer architecture simulation, and cache-conscious software optimizations.

Hill's current research is mostly part of the Wisconsin Multifacet Project that

Work focuses on using the transistor bounty provided by Moore's Law to improve multiprocessor performance, cost, and fault tolerance, while also making these systems easier to design and program.



Gurindar Sohi
Next-Generation Microprocessors

Guri Sohi's research focuses on the design of next-generation microprocessors along with their supporting memory systems. Microprocessor design is currently at an inflection point: the continuously-improving performance of uniprocessors is abating, and microprocessors are turning to multicore or multiprocessor microarchitectures. This calls for new processing models and microarchitectures so that a spectrum of computing applications can run efficiently and reliably on future multicore/multiprocessor chips. Sohi's research group is investigating novel parallel execution models and microarchitectures to enable the concurrent and reliable execution of future computing applications on future microprocessors. Specific research directions include novel execution models for achieving concurrency, new ways of organizing on-chip memory hierarchies, improving software and hardware reliability, and novel hardware/software techniques for managing the resources of chip multiprocessors.



David Wood
Architecture of Multiprocessor Systems

David Wood's main research goals lie in developing cost-effective multiprocessor computer architectures that take advantage of rapidly changing technologies. His research program has two major thrusts: (1) designing innovative architectures to improve future multiprocessors, and (2) developing new tools,

techniques, and workloads to evaluate alternative architectures.

Wood co-directs the Wisconsin Multifacet Project with M. Hill. Multifacet performs research to improve the cost, performance, and reliability of the multiprocessor servers that form the computational infrastructure for the Internet.

A leading thrust of the Multifacet project investigates Log-based Transactional Memory (LogTM). Transactional Memory (TM) simplifies parallel programming by guaranteeing that transactions appear to execute atomically and in isolation. LogTM is a

proposed hardware transactional memory system that extends a directory-based cache coherence protocol to automatically detect conflicting transactions. Unlike earlier TM proposals, LogTM makes commits fast by storing old values to a per-thread log in cacheable virtual memory and storing new values in place. This ongoing, multidisciplinary project is investigating all aspects of transactional memory, including the demands of emerging parallel applications, the interaction with compilers, runtime systems and operating systems, and hardware implementation alternatives.

Condor Project

One current application using Condor is by particle physicists who are searching for evidence of supersymmetry in nature. Data to be collected by the Atlas detector at CERN, shown below, will be compared with data simulated using the Standard Model of the physics of the universe, which is being generated using Condor. To date, Condor has enabled physicists to run simulations that have used more than 215 CPU years in less than two months.

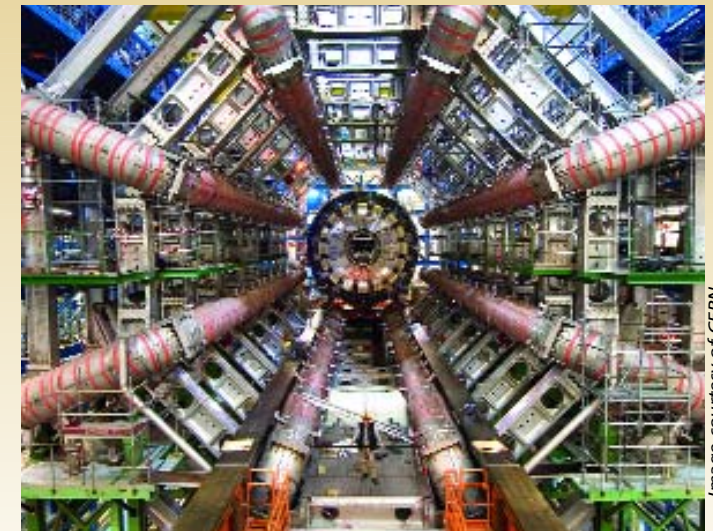


Image courtesy of CERN

"UW-Madison has produced some of the best systems and systems people. Condor is no doubt the most famous product, but the database community is filled with people and ideas created in Madison."

— Jim Gray, Microsoft and ACM Turing Award recipient

graph-based semi-supervised learning algorithms based on spectral graph theory and Gaussian random fields. He is currently exploring semi-supervised

seeks to improve the multiprocessor servers that form the computational infrastructure for Internet Web servers, databases, and other demanding appli-

Computer Graphics and Vision



Charles Dyer

Computer Vision

Automatically analyzing and extracting information from images is of increasing importance as capturing, storing and access to image and video data explodes. Charles Dyer is developing new methods for recognizing faces that combine images of a face from multiple perspectives in order to better cope with occlusion and perspective distortion while facilitating new recognition algorithms. Other work is focusing on using machine learning methods for combining text and images. These projects continue his development of 2D and 3D shape and image representations that lead to new algorithms for analyzing images, including his earlier work on quadtrees, volumetric scene modeling, pyramids, and aspect graphs. In the application area of medical image analysis, Dyer is investigating methods for (1) extracting and reconstructing the 3D structure of white matter fiber tracts from diffusion tensor images; (2) detecting dilated bronchi in CT images of human lungs; and (3) analysis and classification of nuclear sclerosis in the eye from slit-lamp images.



Michael Gleicher

Computer Graphics

The overall goal of Michael Gleicher's research is to create tools that make it easier to create pictures, video, animation, and virtual environments; and to make these visual artifacts more interesting, entertaining, and informative. Current work spans a number of areas within computer graphics. In computer animation, methods are being developed for synthesizing the movement of human

characters. In multimedia, he is exploring the automation of the production of video and images for specific display situations. In scientific visualization, methods are being developed that assist in interpreting large complex objects, such as the vascular network of the brain or the shape and motion of protein molecules.

Database Systems



David DeWitt

Database Systems

One of David DeWitt's current projects, joint with R. Ramakrishnan and J. Shavlik, involves the design and evaluation of scalable techniques for anonymizing large data sets. The challenge is to develop aggregation and generalization techniques that offer simple and flexible ways of releasing data sets to researchers in a way that does not render them useless for conducting research, while insuring the privacy of the individuals contained in the data sets. Techniques are being tested on large cancer data sets that have been collected by the State of Wisconsin Public Health Department.

A second research project involves extensions to the Condor grid scheduling software to provide support for managing large data sets. Within a couple of years even modest cluster nodes will come configured with almost a TB of disk space. Thus, a 1,000 node cluster will have nearly a petabyte of disk space that would sit mostly empty. The Condor DB project is exploring how this space can be effectively used for the long-term storage of input, output, and executable files. Storing user files directly on the nodes of a cluster offers a number of intriguing opportunities, including improved scheduling strategies for jobs, maintaining the provenance of every output file so that a new version can be automatically produced whenever an exe-

cutable is replaced with a new version, and automatically generating new output files whenever a new input file of the appropriate type is added.



AnHai Doan

Efficient Extraction, Integration, and Mining of Data

AnHai Doan's research focuses on information management, with an emphasis on the efficient extraction, integration, and mining of data. Specific problems being investigated are: learning the characteristics of data sources; finding semantic correspondences between source schemas (e.g., do "our-price" at Barnes and Noble and "listed-price" at Amazon.com refer to the same real-world concept?); discovering information on the Internet; developing techniques to efficiently construct and maintain large-scale data integration

"UW-Madison, deservedly well known for its work on databases, computer architecture and distributed computing, is also an international leader in research on performance modeling and program analysis."

— Ken Kennedy, Professor, Rice University

systems; and managing data-rich communities on the Web. Using database and AI (e.g., machine learning) techniques to attack these problems, he builds systems in various application domains (e.g., Web, text) to validate the solutions. The ultimate goals are to transform the Web into a vast collection of structured data sources, and to achieve the widespread use of intelligent systems that efficiently share data across these distributed and heterogeneous sources, with minimal human coaching and supervision.



Jeffrey Naughton

New Technology for Database Management

The ability to store, organize, query, and mine large data sets is of increasing importance in applications ranging from scientific research to business intelligence to personal information repositories and even to the Internet. Jeffrey Naughton's research focuses on developing new technology to facilitate this storage and exploration of data.

Current research interests include data management for large-scale cluster- or grid-based computational environments, and techniques for data storage and query evaluation and optimization in both classical relational systems and new paradigm systems such as XML and stream data management systems. The overall goal is to make systems that manage large amounts of data easier to use, to broaden the types

of data they can manage, to increase the power of their query languages, and to develop new query evaluation algorithms and optimization techniques to enable them to scale to both large and widely distributed data sets.



Raghu Ramakrishnan

Database Systems and Data Mining

Raghu Ramakrishnan's interests are in the areas of database systems, with a focus on data retrieval and integra-

tion, analysis, and mining. He and his group have developed scalable algorithms for clustering, decision-tree construction, and item set counting, and were among the first to investigate mining of continuously evolving and streaming data.

One current project in the area of data mining is called Goal-Oriented Privacy Preservation (with D. DeWitt, J. Shavlik, L. Hanrahan, and A. Trentham-Dietz). Sophisticated mining techniques are necessary tools for understanding complex data sets in an increasing number of application domains. The increasing use of these techniques has also created a heightened awareness of their potential for compromising privacy, and the challenge of controlling data access in accordance with privacy policies while enabling useful analysis has emerged as a central problem. To meet this challenge, this project is investigating the trade-offs between privacy guarantees and the utility of published data for specific analysis tasks.

Networks and Security



Aditya Akella

New Directions in Network Management

As networked systems become more complex, manageability assumes a central role in determining their usability. Today, management features are often "bolted on" to large networked systems, resulting in cumbersome interfaces and limited functionality. Aditya Akella's research focuses on developing new network design paradigms wherein manageability is a central design requirement.

Akella is working on new management approaches for wired and wireless LANs, Internet Service Providers, and private enterprise networks. His research on enterprise network management is centered on the observation that current techniques for enforc-

ing policy in enterprise networks — e.g., firewall rules, router ACLs and VLANs — tend to complicate management and trouble-shooting tasks. He argues that effective enterprise network management is possible only via clean-slate redesign of enterprise networks. Akella is also developing an alternate, backward-compatible, Layer-2 architecture for enterprises that vastly simplifies policy enforcement and eases management tasks.



Suman Banerjee
Wireless and Mobile Networking Systems
Wireless communication is experiencing an unprecedented growth spanning

a wide range of technologies, protocol standards, and spectral bands of operation. Thus, efficient management of network and spectrum resources will become increasingly more challenging.

The WiNGS laboratory, led by Suman Banerjee, is studying resource management problems that arise in the domain of wireless and mobile networking. Questions this research is

“I have considered Wisconsin’s Condor Project to be the leading high throughput computing solution for over a decade, from 1996 when NCSA used Condor to help define the National Technology Grid to today when we are working to create scientific computing technologies of the future.”

— Larry Smarr, Director, California Institute for Telecommunications and Information Technology

addressing include: (1) How should spectrum sharing and coordination be achieved in the unlicensed wireless bands where technologies such as 802.11 wireless LANs, Bluetooth, and ZigBee need to co-exist? (2) How can

under-utilized spectrum resources in the licensed bands be opportunistically harvested by secondary incumbents? In recent work, Banerjee and his students debunked the standard practice of using only non-overlapping channels in 802.11 based wireless LANs. They showed that not only is it possible to exploit partially-overlapped channels, but doing so can lead to significant performance improvements. He also demonstrated that, unlike in wired networks, participation and assistance from clients is crucial in efficiently managing resources in many wireless networks.



Paul Barford
Internet Measurement and Analysis
The Internet’s size, complexity and decentralized structure provide fundamental challenges to effective operation, growth and expansion of capabilities. Paul Barford’s research activities seek answers to questions such as “how can anomalous and malicious traffic be identified quickly and accurately?,” “how can we

automate daily operational tasks?” and “how can we design routers to support future requirements for performance and reliability?”

Research has focused on empirical measurement and evaluation of

Internet systems and behavior. To facilitate this work, the Wisconsin Advanced Internet Laboratory, the largest academic network research lab in the world, has been built. Instrumentation that is deployed in the Internet itself is also managed, enabling a unique perspective on a variety of behaviors, including attacks and intrusions.

Barford’s work has uncovered the complex characteristics of anomalous and malicious traffic in the Internet. This has led to the development of new techniques and systems for identifying and discarding this traffic, with the goal of significantly enhancing the overall security of the Internet. Recent studies on more general aspects of Internet behavior have led to new measurement and visualization tools that enable assessment of key characteristics such as packet loss, which can have a large impact on end-to-end performance.



Cristian Estan
Network Traffic Analysis
One of the main reasons for the success of the Internet is its great flexibility —

people can use it in many ways: read email, browse the Web, or spread viruses and worms. To let users reap the benefits of this flexibility while protecting them from its undesirable consequences, the network must exert some type of control over the traffic it carries. Cristian Estan’s research focuses on building tools that allow us to better understand and control Internet traffic. The two things that make this a challenging task are the difficulty of distinguishing between various classes of legitimate and malicious traffic and the large volumes of traffic handled by today’s networks. To address the first problem he works on traffic analysis and visualization methods that can reveal to a network administrator the

patterns of interest in the current network traffic. To address the second problem he works on improving the throughput of intrusion prevention systems that match network traffic against signatures that describe well-known attacks. Toward this goal, he develops algorithms that improve performance-sensitive modules of the system as well as solutions that use the emerging chip-multiprocessors as a platform for intrusion prevention.



Somesh Jha
Software Security
Shared resources, such as the Internet, have created a global-information infrastructure.

However, shared resources also create risks due to intentional or unintentional malicious behavior. Somesh Jha’s primary research focus is in the area of software security, which tackles problems that arise from vulnerabilities in software.

Jha’s group applies logic-based formal methods to various problems in software security, applying model-checking and program-analysis techniques to address various problems. One project is studying model-based intrusion detection. He has developed context- and environment-sensitive models, which result in low false-negative rates in the context of model-based intrusion detection systems.

A second project is investigating malware detection. A malware detector identifies and contains malware before it can reach a system or network. Jha has developed techniques to test the resilience of malware detectors to obfuscation techniques used by hackers. Malware detectors have been developed that use semantic-aware analysis, i.e., the malware detector analyzes the suspicious program using information about the semantics of each instruction.

Operating Systems & Distributed Systems



Andrea Arpaci-Dusseau
Gray-Box Systems
Large new software systems are not built from scratch, but instead leverage

existing software. One difficulty with using existing software is that it may not always behave as desired — it may have performance, reliability, or security problems. Andrea Arpaci-Dusseau’s research addresses the problem of building layered systems from existing software that cannot be modified. Building layered systems can be simplified by treating each layer as a gray box. In a gray-box system, one starts with basic knowledge of how a layer is likely to be implemented; one then builds successively refined models of the layer by observing how the layer responds to inputs at run-time. Gray-box knowledge allows one both to acquire information about the internal state of an existing layer and to control its behavior in unexpectedly powerful ways.

Arpaci-Dusseau’s group has investigated gray-box systems in three important domains: user-level processes interacting with gray-box commodity operating systems, storage systems interacting with gray-box file systems, and virtual machine monitors interacting with gray-box operating systems. They have developed fingerprinting tools to automatically infer and characterize the behavior of existing layers, and they have shown how gray-box knowledge can be used to improve performance, reliability, and security.



Remzi Arpaci-Dusseau
Smarter Storage Systems

The design and implementation of the next generation of storage systems are the interests of Remzi Arpaci-Dusseau. These systems will harness increases in processing power to deliver improvements in

tems. Storage system innovation has been limited in the past, as such systems are usually built behind a block-based interface, e.g., SCSI. Hence, traditional storage systems have no knowledge of how they are being used, and are limited in the types of functionality they can deliver. To enable a richer set of functionality, he has been exploring the design and implementation of “semantically-smart”

“Wisconsin has some of the best faculty in the world in computer architecture and databases.”

— David Patterson, Professor, University of California, Berkeley

performance, reliability, security, and functionality.

One goal of Arpaci-Dusseau’s research is to enable such advancement within block-level storage sys-

tems. Such a storage system either learns about, or is embedded with, information about the file system above, and thus can implement a much broader range of functions than

a typical storage array, all while exploiting the information available within such a system. For example, a block-level storage array has complete knowledge of the contents of its cache; combining that with semantic file-system level information can lead to improved cache management.



Miron Livny
High Throughput Computing

High throughput computing is a challenging research area in which a wide range of techniques is employed to harness the power of very large collections of computing resources over long time intervals. Miron Livny’s group is engaged in research to develop management and scheduling techniques that empower high throughput computing on local and wide area clusters of distributively-owned resources. Results are incorporated into the Condor system, a widely used high throughput computing system. Researchers from a spectrum of scientific disciplines collaborate in the development and evaluation of Condor. In the area of visual exploration of information, Livny works on developing a framework and tools for intuitive graphical interaction with collections of multimedia data. He works closely with domain scientists on testing and evaluating these tools with real data.



Barton Miller
High Performance Computing and Computer Security

In the area of high performance computing, Bart Miller directs the Paradyn Parallel Performance Tool project, which is investigating performance and instrumentation technologies for parallel and distributed applications and systems. This project investigates research technologies spanning low-level binary code instrumentation to automated performance diagno-

sis to approaches for scaling tools up to large parallel systems.

Miller’s research in the area of binary code instrumentation includes “dynamic instrumentation,” i.e., the ability to analyze a binary program and safely patch code into and out of a running program or OS kernel. In automated performance diagnosis, his group has developed technology to automatically control program instrumentation to collect on-line performance data, evaluate that data, and further instrument the program. In addressing tool scalability, he has developed MRNet, an infrastructure to scale tools to computers with hundreds of thousands of nodes.

Miller’s work in computer security covers the areas of host and network intrusion detection, binary code analysis, and vulnerability assessments. His work emphasizes a combination of theoretical foundations and practical designs and implementations.

ClassAds may be used like access-control lists and certificates, but are more powerful, allowing a wide variety of policies to be expressed. Despite this power, efficient algorithms exist for determining whether access is allowed under the policy specified by ClassAds. There are also algorithms for analyzing policies specified by ClassAds, for example, explaining why a desired access is prohibited or an unintended access is allowed.



Michael Swift
Tackling the Complexity of Commodity Operating Systems

Commodity operating systems support a dizzying variety of hardware and software applications. This has led to an incredible level of complexity in areas that have never been addressed by the research community. For example,

Selected Awards

ACM Doctoral Dissertation Awards to CS Faculty

T. Reps (1983), D. van Melkebeek (1999), A. Doan (2003), B. Liblit (2006)

Fellows of Major Professional Societies

ACM

D. DeWitt (1995), L. Landweber (1995), M. Vernon (1996), J-Y. Cai (2001), B. Miller (2001), R. Ramakrishnan (2001), J. Naughton (2003), G. Sohi (2004), M. Hill (2005), T. Reps (2006), D. Wood (2006)

IEEE

C. Dyer (1998), M. Hill (2000), G. Sohi (2004), D. Wood (2004)

NSF Awards

PRESIDENTIAL YOUNG INVESTIGATORS

D. Joseph (1985), M. Vernon (1985), E. Bach (1986), T. Reps (1986), M. Hill (1989), S. Horwitz (1989), J-Y. Cai (1990), R. Ramakrishnan (1990), M. Ferris (1991), J. Naughton (1991), D. Wood (1991)

CAREER

M. Gleicher (1999), R. Arpaci-Dusseau (2001), A. Arpaci-Dusseau (2002), D. van Melkebeek (2002), P. Barford (2004), A. Doan (2004), S. Jha (2005), C. Estan (2006)

“The University of Wisconsin has a great faculty and an outstanding record of important research contributions in the field of computer architecture.”

— Joel Emer, Intel Fellow



Marvin Solomon
Analyzing Access Policy

Marvin Solomon’s research involves the “ClassAd” (classified advertisement) language, originally introduced in the Condor high throughput computing system. A ClassAd is an associative array mapping attribute names to expressions. ClassAds are used in Condor to match jobs with machines that are available to run them. ClassAds may be used for a variety of other matchmaking tasks in distributed environments. One application is authorization policy in security systems.

Windows XP maintains up to 50 MB of configuration data, and supports more than 35,000 device drivers. Due to the difficulty of managing and testing the OS, this complexity leads to many problems such as increased maintenance cost and decreased reliability.

Michael Swift’s research efforts focus on reducing the complexity of operating system design and operation, thereby leading to increased reliability, security, and manageability. He seeks to understand existing systems and practices and then develop both short-term solutions with low cost and longer-term solutions.

One current project is to improve operating system reliability. Swift identi-

fied device drivers as the primary cause of crashes, and then developed a technique for isolating operating systems

tem, optimized parallel memory architectures, optimal scalable video streaming protocols, and near-optimal, large,

“UW–Madison Computer Architecture is known for over two decades of innovative contributions to commercial computer systems and an outstanding legacy of graduates in academia and industry.”

— Justin Rattner, CTO, Intel

from malfunctioning drivers. This approach can prevent 99% of driver-caused crashes in Linux. In addition, it has a small performance impact and is relatively simple to integrate into existing operating systems.



Mary Vernon
Near-Optimal System Design

Mary Vernon's research develops state-of-the-art analytic performance modeling techniques and applies those techniques to obtain new near-optimal system designs. Efficient analytic models are developed to: (1) identify the bottlenecks in existing system designs, (2) determine the best system performance, (3) develop system designs that (nearly) achieve optimal performance, and (4) achieve near-optimal responses to the input workload during on-line operation.

Vernon's group has developed modeling techniques such as the customized Mean Value Analysis technique, which has proven to be more accurate, provide greater insight into design optimizations, and find solutions up to five orders of magnitude more quickly than detailed simulators for modern systems. Research projects have applied various modeling techniques to achieve a wide range of near-optimal system designs, including commercial bus arbiters, optimal semaphore architecture for a commercial operating sys-

distributed Grid applications.

Optimization & Numerical Analysis



Michael Ferris
Computational Optimization

As systems become increasingly complex, models and tools to determine better ways to design them, more efficient ways to operate them, and techniques to recover from unexpected events or failures, have become more and more necessary. Optimization is a critical component that formalizes ways to look at problems, provides algorithms to find solutions to these problems, and enhances the ability of domain experts to model, operate and improve realistic computer, management, medical, supply chain, engineering or economic systems. Examples from Michael Ferris's research include the design of cancer treatments, traffic tolling, mathematical engines for video-game development, video-on-demand data delivery, economic policy (including GATT and Uruguay Round benefits), coal mine operation, and a variety of problems in structural and mechanical engineering.

Ferris's research is concerned with algorithmic and interface development for large scale problems in optimization, including links to the GAMS and AMPL modeling languages, and software such as PATH, NLPPEC, and FAT-COP. He has worked on applications of

both optimization and complementarity, and has developed tools to allow easy use of grid computing platforms.



Robert Meyer

Large-Scale Combinatorial Optimization
Large and difficult combinatorial optimization problems

arise in parallel computation, computer graphics, database design, communication and supply chain network design, and radiation treatment planning. Robert Meyer's research focuses on the development of mathematical models for these types of application areas, as well as efficient computational techniques for the solution of these large-scale models. For example, beam angle selection (BAS) is an important problem in the treatment of cancer via radiotherapy. BAS requires the determination of a set of "angles" from which radiation will be delivered to a patient by means of a linear accelerator. The data for this problem requires estimates of the amount of radiation that will be delivered from each possible angle to hundreds of thousands of small patient volume elements, and the solution must take into account constraints on the spacing of the beams. Meyer is developing procedures to automate BAS, and thereby generate beam sets that are superior to those that can be obtained from current clinical practice, which relies on expert judgment to obtain approximate solutions to this difficult problem.



Amos Ron

Approximation Theory and Scientific Data Representation
Amos Ron's research is in the area of data representation with

emphasis on several variables. His interests include spline approximations, scattered data approximations, polynomial interpolation, wavelet representations, and Gabor representations.

Application areas include Internet traffic analysis, NMR spectroscopy, medical MRI, molecular biology, and computer graphics.

Within approximation theory, Ron and his colleagues introduced a new class of multivariate, compactly-supported splines on regular grids called exponential box splines, developed a general solution for the multivariate polynomial interpolation problem, and developed the theory of shift-invariant spaces. In the area of data representation, he developed with Z. Shen a general technique called fiberization for analyzing translation-invariant representations. The theory was used to establish a general connection between low-density Gabor representations and high density ones.

In the area of wavelets, Ron and Shen developed the theory of framelets, i.e., a redundant wavelet representation, and devised, together with I. Daubechies, B. Han and Z. Shen, their fast algorithms. He recently introduced general methodologies for

the construction of extremely local wavelet representations in high spatial dimensions.



Stephen Wright

Computational Optimization
Optimization is fundamentally about algorithms for finding the minimizer of

a mathematical function, subject to certain constraints on the variables of that function. A tremendous variety of problems in physical, biological, engineering, and economic systems can be expressed as optimization problems in which the minimizing state reveals important information about the system — for example, the state it attains in nature or its optimal state of operation. Examples from Stephen Wright's research include finding a global atmospheric state most consistent with a set of observations from satellites and ground-based sensors; finding radiotherapy treatment plans for cancer that

WAIL Lab

The Wisconsin Advanced Internet Laboratory is a one-of-its-kind facility for conducting networking, security and distributed systems research, founded by a grant from Cisco Systems and the TOSA Foundation.



deliver specific doses of radiation to the tumor while minimizing exposure of surrounding healthy tissue; and controlling the operation of a chemical or manufacturing process to maximize its efficiency or throughput.

Wright's research includes the design, analysis, and implementation of algorithms, and the use of grid computing platforms to solve very large problems. Additionally, he works on interdisciplinary projects with domain scientists and engineers involving applications of optimization to their disciplines.

Programming Languages & Software Engineering



Charles Fischer
Compiler Support for Multicore and Multithreaded Processors
Charles Fischer's research interests

focus on compiler design and implementation with recent focus on exploiting new architectural capabilities. A number of emerging computer architectures are multicore or multithreaded, supporting efficient simultaneous execution of several program threads. But where are such threads to be found? Most programs are single-threaded, and extracting a useful number of independent threads is difficult.

Fischer proposed creating assist threads that concurrently assist an application program as it executes. Assist threads can perform many functions. They can monitor a program's execution, looking for errors, inconsistencies, or hotspots. They can anticipate a program's flow of control, prefetching instructions or data. They can "shadow" a program's execution, performing garbage collection or optimizing code already executed.

Current research involves ways of automatically finding and generating

assist threads, as well as evaluating their effectiveness in enhancing program performance. Use of type information in object-oriented languages is a simple and effective way of identifying non-interfering method executions, supporting method-level parallelism in otherwise sequential programs.



Susan Horwitz
Tools to Make Programming Easier and Less Error-Prone
Susan Horwitz's research focuses on new tools and techniques

to help programmers write better code. This involves addressing problems such as how to test code more thoroughly with less effort; how to find the errors when a program doesn't work correctly; how to reorganize old code so that it is easier to understand and to modify; how to help programmers understand how existing code works, and how it would be affected by proposed modifications; and how to find vulnerabilities to malicious attacks in existing code, and preventing such attacks at run time.

A novel aspect of Horwitz's work is the use of a program representation called a Program Dependence Graph (PDG), and an operation on PDGs called program slicing. The backward slice from a statement in a program includes just the code that might affect the execution of that statement; this makes slicing very useful in applications like debugging, refactoring, merging program versions, and clone detection. Her work has included both identifying new uses for slicing and improving slicing algorithms (to be more efficient, more accurate, and to apply to more language constructs).



Ben Liblit
Cooperative, Statistical Debugging
Computer technology is rapidly perme-

ating all spheres of society. When software fails, more and more people are inconvenienced or harmed. Yet highly feature-rich and adaptive systems make it impossible to anticipate, test, or even understand all possible real-world configurations before deployment. Meanwhile, a technology-hungry public constantly clamors for the next new thing, forcing software vendors to make difficult choices between quality and time-to-market.

Ben Liblit's research concerns software bugs: why they happen, how to prevent them, and how to rapidly diagnose and fix them when they occur. His approach uses a combination of static program analysis, dynamic instrumentation, and statistical machine learning. A main focus is on problems that arise in widely deployed software systems. Lightweight instrumentation with sparse random sampling lets him collect small amounts of feedback across a large user community. Statistical models are then used to identify program behaviors that are strongly predictive of failure, and both static and statistical methods to track these (mis)behaviors back to the root causes of bugs. By using this approach, called "Cooperative Bug Isolation," developers can turn their users into a massive distributed bug-hunting resource that helps make the software better for everyone.



Thomas Reps
Advanced Static-Analysis Techniques for Assuring Reliable Software
Thomas Reps's research focuses on

methods to help programmers develop correct, reliable, and secure software. The goal is to create better tools for manipulating programs and analyzing their execution properties.

Current work on static analysis via 3-valued logic addresses the problem of analyzing programs that use dynamic allocation and freeing of storage cells

and destructive updating of structure fields — features found in all modern programming languages. His techniques provide a way to create finite-sized descriptors of memory configurations that abstract away certain details, but retain enough information so that the analysis can identify interesting properties.

In Reps' work on analyzing stripped executables, methods have been devised that — without debugging information — identify variables, data objects, and types, and track how they

"Thirty years of seminal contributions to the database field."

— Michael Stonebraker, Professor, University of California, Berkeley

are manipulated by the executable. This information provides answers to such questions as "what could this dereference operation access?" and "what function could be called at this indirect call site?," which can help human security analysts understand the workings of COTS components, plugins, mobile code, and DLLs, as well as memory snapshots of worms and virus-infected code.

Theory of Computing



Eric Bach
Algorithmic Number Theory
The integers are simultaneously the simplest and most complicated of

mathematical systems. Eric Bach's research deals with the computational aspects of this subject. More precisely, he is interested in the design and analysis of algorithms for problems in number theory and related algebraic areas. These algorithms have many applications, from computer design to reliable and secure communication

(e.g., encryption uses number-theoretic principles) to computationally intensive scientific applications (e.g., computer algebra requires exact arithmetic).

Results include the first efficient algorithm to generate random integers with known factorization, estimates for the reliability of randomized algorithms using pseudo-random generators, and an accurate heuristic model for generators of multiplicative groups. If this model is correct, n can be tested for primality using pseudo-prime tests on the first $O(\log n)$ primes. This is currently

conjectured to be the fastest prime test.

Recently, Bach has studied methods for accurately computing sums over primes, solution probabilities for systems of quadratic polynomials over finite fields, and arithmetic and combinatorial properties of quantum random walks.



Jin-Yi Cai
Computational Complexity Theory
Jin-Yi Cai works in the area of computational complexity theory — the study

of quantitative laws that govern the limitation and power of computation. The primary objects of study are complexity classes. There are many open problems in this area, the most famous of which is the $P \neq NP$ conjecture.

In core complexity theory, Cai proved the probability one separation of the Polynomial Time Hierarchy from PSPACE by a random oracle. He also proved the Hartmanis Conjecture on sparse sets for the complexity classes P and NL, and the unconditional containment of the class S_2^P in ZPP^{NP}.

Currently his group is interested in matchgate computations and holo-

graphic algorithms. They have shown an equivalence of expressive power of general matchcircuits and planar matchgrids. A common theme in Cai's work is the application of non-trivial mathematical tools to the study of algorithmic problems with the goal of better understanding computation at a fundamental level.

"The Computer Sciences Department has produced an impressive series of contributions, especially in the areas of database systems and data mining. What has been particularly impressive is the rigorous scientific work that is coupled with serious systems implementation and evaluation on real large-scale systems. This truly sets UW-Madison apart in their contributions to the scientific community."

— Usama Fayyad, Chief Data Officer and Senior VP, Yahoo!



Shuchi Chawla

Approximate Combinatorial Optimization

Designing approximation algorithms for NP-hard opti-

mization problems is the focus of Shuchi Chawla's research. This includes traditional combinatorial optimization problems, as well as the newer areas of multi-stage stochastic optimization and algorithmic mechanism design.

Most real-world optimization problems involve uncertainty in their inputs or constraints. For example, in driving from home to work, the best route to take may depend on the current traffic conditions. A solution is not just a single route, but a complex decision diagram for which directions to take depending on the observed traffic conditions. Can a simple, approximate decision diagram be obtained with known expected performance? Given an algorithm for the deterministic shortest paths problem, how can it be converted to one for the stochastic version? These are the kinds of questions

she is studying for various stochastic optimization problems.

Optimization problems in distributed settings often involve the participation of several self-interested agents, each of whom can influence the result to her own benefit by controlling her input appropriately. For such mechanism design problems, the goal is to

devise techniques for giving incentives to the selfish participants so that they follow the protocol of the algorithm while achieving the desired outcome. Chawla is studying several such problems from an algorithmic perspective.



Deborah Joseph
Complexity Theory and Algorithm Design

Deborah Joseph's research in computational biology is focused on computational tools for genome sequencing and analysis. This includes the development of dynamic data structures and algorithms for fragment assembly, the development of algorithmic techniques for identifying repetitive sequences, and the utilization of graph theoretic methods for rapid homology detection in the analysis of anonymous sequences.

A second area of research interest is in complexity theory. Nearly a half century of research has yielded a plethora of complexity classes, and yet major

questions remain open: Does $P = NP$? Does $P = Pspace$? Joseph is investigating the structural properties of sets in these classes and proof techniques for separating and collapsing complexity classes.



Dieter van Melkebeek

Computational Complexity

Computational complexity studies the inherent difficulty of computational problems, measured by the amount of resources it takes to solve them: How do the required time and space grow as a function of the input size for various models of computation such as deterministic, randomized, and quantum machines? The issue is of paramount importance for NP-complete problems because of their preponderance in science and engineering.

One of Dieter van Melkebeek's recent research goals is concrete lower bounds for NP-complete problems. Although problems like satisfiability are conjectured to require exponential time on any feasible model of computation, the existence of a deterministic linear-time algorithm is not yet ruled out. In the case where the algorithm only uses a subpolynomial amount of space, van Melkebeek has shown a time lower bound of n^ϕ where ϕ denotes the golden ratio.

Another research project deals with the question: Does somewhat more time allow us to solve strictly more problems? Although time hierarchies for deterministic machines formed the cradle of computational complexity, such results have remained elusive for semantic models like randomized and quantum machines with bounded error. Van Melkebeek recently proved a time hierarchy that applies to virtually any semantic model with one bit of advice.



Printed August, 2006.

Designed, written and produced by University Communications for the Department of Computer Sciences. Charles Dyer, editor.

Photography: Bob Rashid, cover images, pages 2, and 8-27; University Communications, pages 1 (aerial), 6, and 36; UW Archives, pages 1 (computer), 4, and 5.

©2006 Board of Regents of the University of Wisconsin System

The University of Wisconsin-Madison is an equal opportunity and affirmative action educator and employer.



THE UNIVERSITY
of
WISCONSIN
MADISON

Department of Computer Sciences
1210 West Dayton Street
Madison, WI 53706-1685
608/262-1204
www.cs.wisc.edu

2006