The Computer Science Department began in 1964 with three faculty and has been growing ever since—building on our strengths in machine learning, computer vision, information retrieval, software engineering, networking, multiagent systems, and cognitive science, and expanding into new fields like bioinformatics, graphics, distributed systems, security, and databases.

Since 2002, we have grown dramatically, adding eleven new tenure-track faculty. Given our incredible growth, we are highlighting many of our young faculty in this booklet. We are now 43 faculty strong—promising vitality and innovation in emerging fields for years to come. And so this is a good time to inform our colleagues, collaborators, future students, and alumni about all the exciting things going on in the department.

The cover image is a graph of connected individuals and a shortest path route. This visualization is based on some breakthrough work by one of our new faculty in data mining (read more about it in “Graduate students discover it’s a small, small world”). But we also chose this cover to emphasize what we believe is one of our strongest characteristics—the connectivity among our faculty and our commitment to productive research and teaching through collaborative work.

Indeed, the department, which is part of the College of Natural Sciences and Mathematics, is remarkably committed to interdisciplinary research, with unusually large intra-departmental cross-field research, often funded through major interdisciplinary grants. Our annual research funding exceeded $12.5 million in 2005, coming from private industry and all levels of government. The department maintains significant research collaborations with more than 30 industry-leading technology companies.

In 2006, the department celebrated 41 years of graduate training. As one of the highest ranked and highest impact graduate research programs in the nation, UMass Amherst CS is also one of the most competitive. Currently, 230 graduate students, 333 undergraduate majors, and 1,550 classroom students enjoy a broad and comprehensive teaching and mentoring program that provides a solid foundation for a lifetime of learning.

We hope you enjoy catching up on our activities or learning about us for the first time. If you’d like to find out more, we’ll be happy to hear from you. Contact information and department details can be found on the last page. We also hope that you will pass this booklet on to colleagues and students who might be interested in Computer Science at UMass Amherst.
Graduate students discover it’s a small, small world

Ozgür Şimşek’s route to computer science was a multi-hop path that took her from Turkey to the U.S. to study industrial engineering, to a contract research firm, and finally to UMass Amherst. Jennifer Neville also landed on campus through unconventional means after sampling math, physics, and cognitive science, getting married, starting a family, relocating a number of times, and finally discovering her true calling in computer science.

Ozgür (left) and Jen (right) both began the graduate Computer Science program in 2000 and built a support network based on their similar backgrounds and interests. In graph theory parlance this is homophily—the tendency of like to associate with like. And now their recent works in graph-based relational data mining, with Associate Professor David Jensen, reflect that same property observed within many complex networks.

Ozgür recently delved into the classic small-world phenomenon, first proposed by Stanley Milgram in 1967, in order to solve an open question about navigating social networks. Given a network of individuals in which edges connect direct acquaintances, it has been known that short paths between people are typically due to two kinds of intermediaries—those with similar characteristics to the pair in question (nodes with high homophily) and those who have many friends (nodes of high degree).

Ozgür reasoned that one could exploit both characteristics simultaneously to efficiently identify a short path—even when the full connectivity of the graph is unknown. She developed a probabilistic approximation called expected-value navigation that finds paths remarkably close to the optimal path—a finding offering insight into the oft-quoted “six degrees of separation.”

Jen has also been exploring social networks, but from a classification perspective. In domains such as bioinformatics, citation analysis, epidemiology, finance, and fraud detection, the connections among entities are often crucially important to learning accurate classification models. Recently, in collaboration with the National Association of Securities Dealers, Jen set out to identify fraudulent stockbrokers.

Her method involves mining relational information about brokers’ current and former business connections, brokerage firms, and the public disclosures about complaints, legal actions, and the like. Fraud and malfeasance are often social phenomena, communicated and encouraged by the presence of other individuals who also wish to commit fraud. Jen therefore reasoned that fraud could be detected by examining the broker’s social network.

However, building a classifier to predict likely fraud is hard because of the enormous feature space and dependencies. To tackle
the problem, Jen developed a classification technique called relational dependency networks. The algorithm automatically constructs and searches over aggregated relational features that summarize information from the social network, with a feature selection process that reasons accurately about the dependencies among entities.

Then Jen applied the method to discover relational fraud patterns in the brokerage data. The models automatically uncovered a statistical relationship of fraudulent activity among brokers working at the same branch, thus demonstrating the potential for significant improvement in fraud detection using social network analysis.

Özgür describes her experience in the department as great fun and “a collaborative environment in which people interact a lot. The social network analysis research is completely outside the scope of my dissertation in reinforcement learning, but it sprang out of serendipitous discussions.” Jen agrees and adds, “Although the department provides a very challenging academic environment, it also offers a supportive atmosphere where students and faculty are consistently encouraged to excel.” Inspired by their successes at UMass Amherst, Jen and Özgür are pursuing careers in computer science. Jen is now an assistant professor at Purdue University and Özgür hopes to join the faculty ranks within the year.

The graduate program

The Computer Science Department has offered graduate education for more than forty years, granting its first M.S. degree in 1967 and its first Ph.D. in 1974. Today, it is home to more than 200 graduate students from many different backgrounds and all parts of the globe. Admission is very competitive: a typical entering class numbers between 20 and 30 students selected from about a thousand national and international applicants. Students are involved in research programs from the beginning of their graduate careers. By the time they graduate, many have multiple published papers to their credit.

Once students enter graduate studies, the department aims to see that they not only successfully complete a comprehensive academic program but also prepare themselves well for their future careers. To this end, faculty encourage doctoral students to develop their professional credentials and skills before they enter the job market. Everyone has to build up a portfolio much like a professional’s curriculum vitae, including the student’s experience (courses and grades), accomplishments (papers and reports published), and service. The portfolio also contains letters of recommendation from faculty and an example of the student’s research work in the form of a “synthesis project.”

This project entails working with two faculty from different disciplines, underlining the department’s fundamental philosophy of graduate education and research: that interdisciplinary breadth is both a rich source of excellent research ideas and a prerequisite for staying abreast of ideas in computer science itself. Most synthesis projects lead to published papers and frequently to Ph.D. dissertations. Students are also encouraged to incorporate breadth into their graduate programs through a diverse set of course requirements.

Upon graduation, students consistently find positions in some of the best academic and industrial settings. More than half of graduating Ph.D.s hold positions at top-ranked institutions. For example, recent graduates have joined MIT, Carnegie Mellon University, and the University of Edinburgh, Scotland. Of these academics, 35 percent have received the prestigious NSF CAREER award. Another 16 percent of graduates work in industry research labs such as AT&T, Google, Hewlett Packard, Intel, Microsoft, SRI and PARC.
Professor Robert Moll epitomizes the Computer Science Department’s commitment to undergraduate teaching and engaging students in the joy of learning. Since 2001, he has served as associate chair for academic programs after a nine-year stint as undergraduate program director. In these roles, Robbie has taught and advised thousands of computer science students, making him one of the department’s most recognizable faces.

Computer science students reach the department by different routes—including direct admits, transfers from state and community colleges, and transfers from other majors. Their backgrounds and preparation vary widely, so Robbie has worked to ease their transition into a demanding program by ensuring that early courses provide a solid foundation for success for all students.

For example, the new Information Technology Minor program has brought many students with no programming experience into computer science courses. These IT students and a number of pre-majors have to compete with very experienced majors, and at first they did poorly as a group. So Robbie redesigned the introductory courses, creating one for students with programming skills and another for those he calls “raw beginners.”

For extra help, Robbie adapted the Online Web-based Learning (OWL) system developed on campus to automatically grade Java programs and provide personalized tutoring. He also used an Evaluator-Integrated Book (EIB) to provide an online textbook-style narrative to accompany the automated homework analysis. With OWL and EIB, 86 percent of students passed the final exam, the vast majority of whom had failed in the original configuration. Notably, 61 percent of students now said they would consider taking more computer science courses and with the number of women in the new course quadrupling, it has become an important recruiting tool for women and other underrepresented groups.

Robbie also has an impact on more advanced courses. After a curriculum reorganization several years ago, he designed and developed the course Programming Language Paradigms that follows the introductory courses. By introducing four fundamental programming paradigms—functional, logic, imperative, and object-oriented—the course teaches students conceptual analysis, relations, state dependency, and object behavior, which prepares them for an array of courses that follow.

The best mathematics and computer science students also benefit from his popular combinatorics course, which draws on his research on artificial intelligence methods applied to combinatorial optimization. He has also co-authored five computer science books.
textbooks, including two introductory Pascal programming texts, and has been nominated four times for the university’s Distinguished Teaching Award.

Robbie finds undergraduate courses “a pile of fun to teach.” Thinking of teaching as a variant of stand-up comedy, he adds, “I’m mostly not funny, but bringing a comic’s sensibility to the front of class makes it more fun for me, and biases my delivery toward a ‘punch-line’ style, which helps all of the attention-limited students out there.” Two principles dominate Robbie’s view of teaching: “classical virtues” and the “social process” of learning mathematics and computer science.

“Students have such an erratic sense of trustworthiness, independence, determination, reliability, hard work, and so forth, that I feel that it’s my duty to push these principles at every opportunity,” he says. As to learning, “I think I’ve said a million times over the last few decades that if you can explain a program or a proof to someone at an imaginary cocktail party in a minute or two, then, and maybe only then, have you understood something.” These concepts ring true for Irene Ros, a recent graduate and best software engineering student award winner who says, “I am so happy that my first CS course was taught by Robbie Moll—he made the material fun and stimulating.”

Award-winning teachers

The department’s dedication to undergraduate teaching is reflected in the many awards faculty have received over the years. Professors David Mix Barrington, Brian Levine, Prashant Shenoy, Emery Berger, Ramesh Sitaraman, James Allan, J. Eliot B. Moss, Jim Kurose and Andrew McCallum have all been holders of Lilly Teaching Fellowships from a competitive awards program established in 1986 to cultivate teaching excellence.

Professors Roderic Grupen, Victor Lesser, Jim Kurose, and Arnold Rosenberg have been winners of the College of Natural Sciences and Mathematics Outstanding Teacher Award; Professors Sridhar Mahadevan and Lee Osterweil won similar awards at Michigan State University and the University of Colorado respectively.

Jim Kurose is also an eight-time recipient of the Outstanding Teacher Award from the National Technical University, winner of the 1996 Outstanding Teacher Award of the Northeast Association of Graduate Schools and of the IEEE Taylor Booth Education Medal. With former Dean of the College of Engineering Joseph Goldstein, he founded the Commonwealth Information Technology Initiative (CITI), for which they received a Public Service Award from the University of Massachusetts President’s Office in 2002. Now directed by Joe and Professor Rick Adrion, CITI aims to improve computer science education across the 28-campus system of Massachusetts public higher education.
Neuroscience takes a leaf from machine learning book

For almost three decades, Professor Andrew Barto has studied learning in both machines and animals. His pioneering work has made reinforcement learning a major component of modern machine learning research, and his 1998 book *Reinforcement Learning*, co-authored with his first Ph.D. student, Richard Sutton, remains one of the top-cited books in computer science. In short, he’s a computer scientist to the core but one with an abiding interest in the psychology and neuroscience of animal learning.

So it is personally rewarding that his research group developed an algorithm—Temporal Difference (TD)—that has become a leading model of dopamine neuron activity in the brain. First appearing in Sutton’s 1984 Ph.D. dissertation, the TD algorithm has been the basis of much research by both machine learning researchers and neuroscientists around the world.

The brain’s dopamine system has long been linked to reward learning and reward-related behavior. Dopamine neurons appear to be part of a system that learns to anticipate upcoming rewards. Specifically, they “learn” to produce bursts of electrical activity whenever the animal unexpectedly senses a signal that it is about to receive a reward. If, however, the signal is expected, the burst does not occur. As a result, a rat’s dopamine neurons “learn” to fire at the earliest clear sign that it will shortly receive a morsel of food.

Inspired by the likeness of the dopamine system to the TD algorithm, Andy described the algorithm to neuroscientists in the mid-1990s. Since then, other researchers, both neuroscience theorists and experimentalists, have developed the connection in many interesting directions. Although discrepancies between the two systems have emerged, the view that the brain’s dopamine system operates on the principles of reward prediction as laid out in the computational theory of reinforcement learning from Andy’s group has become the “standard”—though not entirely uncontroversial—view of dopamine neuronal activity.

“The TD algorithm did not result from an effort to model the activity of dopamine neurons,” says Andy. “It was instead the result of trying to solve a computational problem. But the correspondence exists because nature has had to solve this problem as well.”

The dopamine system is, however, much more complicated than any existing model. As one example, dopamine neurons respond to many stimuli that do not signal reward—to unexpected changes in lighting or a sound, for example. Andy reasons that such activity “makes sense from a computational perspective if stimuli like these act as temporary rewards to drive exploratory activity; in other words, if they were the basis of curiosity.”
Andy and Professor Satinder Singh of the University of Michigan recently conducted a computational study to explore “intrinsically motivated” reinforcement learning, in which unexpected salient events generate rewards. They simulated a simple “playroom” in which various items (a light switch, ball, bell, etc.) could be moved or activated by an “agent.” Even though the agent receives a “real” reward only at the end, it generates for itself “intrinsic reward” when something it does produces an unexpected change in the state of the playroom (such as when the lights come on unexpectedly).

“Our goal,” says Andy, “is to show that intrinsically motivated reinforcement learning is a route to a truly open-ended form of machine learning. And maybe we’ll learn something about the brain, too.”

Sridhar Mahadevan’s research addresses a longstanding puzzle in AI: how can agents transform their temporal experience into multiscale task-independent representations that can effectively guide long-term task-specific behavior? In other words, how can experience be generalized so it can be applied to new problems in the future?

Since the 1950s when Arthur Samuel at IBM developed a program that learned to play checkers, research in agent learning has typically assumed that a human designer hand-codes a representation of the state space—for example, Samuel used a polynomial function of board features to approximate the value of each state in checkers. In the 1960s, Saul Amarel advocated designing agents that learn representations based on global state space analysis, but subsequent work did not show how state space analysis could lead to a generative theory of function approximation.

Now Sridhar has developed a novel manifold-based framework unifying Samuel and Amarel’s paradigms of behavior and representation learning. Algorithmically, the framework comprises four phases. Initially, agents learn a discrete manifold representation of a given environment, which can be viewed as a discrete topological graph representing the states reachable through single or multi-step actions. Next, the graph is analyzed using spectral clustering techniques to reveal “bottlenecks,” symmetries, and other geometric invariants.

In the third phase, an orthonormal set of task-independent basis functions called proto-value functions are extracted from the environment’s topology: these basis functions capture large-scale geometric invariants to which all value functions on the state space must adhere. In the final phase, proto-value functions are combined with rewards to approximate task-specific value functions.

The theoretical basis for proto-value functions draws upon links between discrete and continuous mathematics: Riemannian manifolds and the spectral theory of graphs, heat diffusion equations, and Fourier and wavelet analysis on graphs. Whereas previous methods required cubic time, Sridhar and Mauro Maggioni of Yale’s Department of Mathematics recently showed how proto-value functions could solve the well-known Bellman equation for an important class of “diffusion” problems in approximately linear time.

Proto-value functions

Andy and Professor Satinder Singh of the University of Michigan recently conducted a computational study to explore “intrinsically motivated” reinforcement learning, in which unexpected salient events generate rewards. They simulated a simple “playroom” in which various items (a light switch, ball, bell, etc.) could be moved or activated by an “agent.” Even though the agent receives a “real” reward only at the end, it generates for itself “intrinsic reward” when something it does produces an unexpected change in the state of the playroom (such as when the lights come on unexpectedly).

“Our goal,” says Andy, “is to show that intrinsically motivated reinforcement learning is a route to a truly open-ended form of machine learning. And maybe we’ll learn something about the brain, too.”
Using software engineering to reduce medical errors

Preventable medical errors cause at least 97,000 deaths each year in the U.S. That’s equivalent to the crash of one fully loaded jumbo jet every day. Professors Lee Osterweil, Lori Clarke, and George Avrunin want to reduce this toll. And the technologies they have developed for defining software processes and analyzing software systems are proving remarkably useful in several medical settings.

Through a National Science Foundation (NSF) grant, the researchers collaborate with Beth Henneman, an assistant professor of nursing on campus and Phil Henneman, director of the Emergency Division at Baystate Medical Center in Springfield. “It’s been gratifying to see how readily and enthusiastically Beth has picked up on our process definition language and made good use of it,” says Lee.

Beth Henneman concurs: “This project has completely changed the way I think about my work and is transforming the way I teach nursing.”

She’s been working closely with the Computer Science team to use their Little-JIL process language to define blood transfusion processes and expose potential vulnerabilities while doing so. Once it has defined a medical process, the Propel property elicitation system precisely represents the applicable medical protocols. Then the FLAVERS finite-state verification system determines if any way of “executing” the process will violate any of the medical protocols. This and other types of analyses highlight vulnerabilities that medical professionals can attempt to rectify—and then use the software to evaluate the modified process.

The UMass Amherst team also works with Phil Henneman to use process definition and various forms of analysis to streamline the flow of patients through his Emergency Room. The goals are to increase capacity, reduce patient waiting times, and improve quality of care. For example, graduate student David Miller is simulating different scheduling systems to predict which protocol will minimize patient wait time (one simulation is shown in the figure).

Detecting vulnerabilities in chemotherapy processes is the aim of the team’s collaboration with Dr. Wilson Mertens, medical director of Baystate’s D’Amour Cancer Center. Dr. Mertens shares the enthusiasm about the team’s approach, saying: “This work has the potential to completely transform the way cancer care is delivered in the United States.”

Principal Investigator Lori Clarke notes an additional benefit: “This application area is exposing shortcomings of our technology and leading to important new directions for our research.” As computer science becomes integral to twenty-first-century medical care, the UMass Amherst team is pioneering a promising new application while also advancing their basic software engineering research.

Pictured at the D’Amour Cancer Center are (from left): Drs. Lucinda Cassells and Wilson Mertens, George Avrunin, Lori Clarke, graduate student Rachel Cobleigh, Lee Osterweil, undergraduate Irene Ros, and graduate student Mohammed Raunak.
What machines can teach us about molecules

Robots have long been important in manufacturing—and they are now becoming part of our everyday lives. Roombas, for example, vacuum the floors in millions of homes and their colleagues mow lawns and clean swimming pools without human supervision. However, these robots still lack a fundamental human skill: the ability to manipulate objects with dexterity.

Assistant Professor Oliver Brock (pictured) and Professor Roderic Grupen in the Computer Science Department’s robotics laboratory believe that robots have to learn such dexterous manipulation skills by interacting with their environment. Their work in autonomous mobile manipulation combines mechanisms, sensors, and algorithms that one day will enable robots to build habitats on Mars, service satellites in orbit or assist the elderly in their homes with daily chores. Currently, their robots—Dexter and uMan—rank among the most capable manipulation platforms in the field.

Surprisingly, some of the insights developed in the context of robotics are proving useful
in the seemingly unrelated field of structural molecular biology. Oliver, along with other colleagues in his field, have come to realize that proteins, the organic machines inside every living cell that perform a variety of important biological functions, can be viewed as tiny robots. Each of these proteins consists of a sequence of amino acids. The protein folds into a three-dimensional shape that is unique for each amino acid sequence. Based on this shape, the protein performs its function by interacting with other molecules inside the cell. During these interactions, the protein often has to change its shape—to bind to another protein, for example.

The human genome project has decoded all of the amino acid sequences that describe the proteins produced by the human body. However, knowledge of this sequence is not very useful in itself. Only knowledge of the three-dimensional structure of proteins can directly reveal their function and therefore shed light on the complex processes that govern life—and provide clues to finding cures for many diseases. The prediction of the three-dimensional structure of a protein from the sequence of amino acids is known as the protein structure prediction problem, one of the grand challenges in molecular biology.

Oliver’s research in molecular biology applies insights from robot motion planning to protein structure prediction. In spite of the vastly different scale of robots and proteins, both of these problems amount to a very similar computational challenge requiring algorithms that can efficiently build a representation of a very high-dimensional space. The algorithms developed by Oliver and his students in the context of robot motion planning have proven to be highly effective for this purpose. Running on a single personal computer they are able to match the performance of state-of-the-art algorithms that run on tens of thousands of computers in parallel.

But even if the protein structure prediction problem is solved, the function of many proteins will remain elusive unless their flexibilities are known. These flexible regions allow changes in the protein structure when a protein binds with other molecules. To understand how these changes occur, Oliver and his students rely on well-known techniques from robot kinematics. These methods, for the first time, permit the full consideration of protein flexibility, something that so far has been considered computationally intractable.

A solution to the protein structure prediction problem and the ability to understand the interactions among proteins using computational methods could radically change the face of medicine. Finding cures for diseases might become a question of running a computer program, rather than of conducting hundreds of expensive and time-consuming laboratory experiments and clinical trials. And maybe, one day, we can say that robots helped us to achieve this goal. ■
Goodbye to gridlock

For several years, Distinguished Professor Arnold Rosenberg has been working with various collaborators to develop a theoretical foundation for the problem of scheduling large computational jobs with complex inter-task dependencies on Internet-based computing platforms (so-called grid computing).

What makes grid platforms particularly challenging to use efficiently is the unpredictability of the remote “workers,” which precludes the standard strategies, such as critical path analysis, that were developed for scheduling complex computations on conventional multiprocessor platforms. This “temporal unpredictability” is largely a product of an unreliable network and lack of worker commitment—workers seldom guarantee resources, so work is performed at an unpredictable rate. So what is the owner of the large, complex job to do in an Internet-based computing environment?

Contemplating this problem led Arny to suggest a new scheduling paradigm for complex computations with interdependent constituent tasks. “We cannot control the environment in which we are performing the computations, but we can exercise some control over the way we interface with that environment,” he explains.

So he proposed to strive for schedules that allocate/execute a complex computation’s tasks in a manner that always maximizes the number of tasks eligible for allocation to remote workers. The hope is to minimize the likelihood of idle workers, a frequent occurrence in other scheduling methods due to “gridlock” stemming from all tasks having been assigned to other workers.

The goal of the proposed new paradigm is so ambitious—maximizing the number of eligible tasks at every step of the computation—that it’s no surprise that many computations admit no optimal schedules. However, large classes of scientific-type computations, including broad classes of wavefront-oriented and convolution computations, do admit optimal schedules under this new paradigm.

A much more exciting development, though, is a foundation for an algorithmic scheduling theory that crafts optimal schedules for large classes of ad hoc computations. The algorithms in this setting decompose a given computation, whenever possible, into a simple set of “building blocks,” and, by scheduling them optimally, “reassemble” the initial computation in an optimal order. Arny continues to develop this foundation by extending its application to broader classes of computational problems.

Just as exciting as the emerging theory are ongoing simulation experiments that attempt to quantify the computational impact of the new scheduling paradigm in Internet-based computing. Simulated scheduling of real scientific computations under different conditions have shown consistent gains for Arny’s new paradigm compared to competitive methods currently employed in grid environments. Observed gains in expected computation speed varied from a few percent for so-called expansive computations—notoriously easy to schedule well—to gains of 15 to 20 percent for most classes of computations, including convolutional and “reductive” ones. ■
Universal teacher

The best teachers provide creative, consistent feedback, identify topics confusing to their students, and tailor explanations and challenges to help overcome obstacles to learning. In an ideal classroom, a mentor would help each student speed through easily grasped topics while carefully addressing points of difficulty and never losing patience or moving beyond the appropriate level until the student had mastered the goals.

Such a classroom can exist, even with student-teacher ratios rising nationwide. Computer Science faculty Beverly Park Woolf, Andrew Barto, and Sridhar Mahadevan, research associate Dr. Ivon Arroyo, and Professor of Mechanical Engineering Donald Fisher are building sophisticated, web-based, intelligent tutors that understand students’ learning needs, optimize presentation of materials, and demonstrate effective use of tutoring strategies. Intelligent tutors, used in traditional classrooms with caring teachers, offer the advantages of individualized instruction without added personnel costs.

Drawing on new cognitive research on human learning and educational theories of teaching, this work uses techniques from social and behavioral sciences, linguistics, education, and psychology. Machine learning algorithms customize the choice of hints for individual students based on their cognitive profile, gender, spatial ability, and math fact retrieval speed. Hierarchical, probabilistic models enhance the tutor’s ability to assess the student’s state of mastery. Latent variables, such as a student’s prior knowledge and level of engagement in the tutoring process, are represented. The tutors make decisions by evaluating both a student’s speed of learning and probable retention.

Bev (pictured with Computer Science graduate student Toby Dragon) has collaborated with colleagues from chemistry, biology, ecology, geology, and medicine (to create a tutor for advanced cardiac life support). A particularly exciting example, developed with partners in the Psychology Department, tutors elementary and high school mathematics. Students logged on to this tutor are challenged with geometry problems set in the rainforests of Borneo, where knowledge of mathematics is required to guide them safely through a virtual multimedia adventure.

Using these tutors in classrooms highlights effective teaching interventions and ways to tackle complex topics and student differences. Large-scale experiments in schools help determine the practical significance of each enhancement and have produced a measurable impact in classrooms. For example, on state-required exams, students who used these tutors for only two or three hours improved their scores by 10 to 12 percent. Bev and her team, working with colleagues in the School of Education, are continuing to develop teaching tools on the cutting edge of technology. New interests include collaborative learning and work with disabled students. With computers gaining speed and agility in understanding how to teach, and behavioral science and psychology broadening our knowledge of how people learn, the vision of a “teacher” for every student has become a real possibility.
Measuring network performance with tomography

It’s easy to see why it’s becoming ever more difficult to model and measure the performance of the Internet, with its thousands of autonomous networks, hundreds of thousands of routers, and hundreds of millions of end users. Router and link capacities are also increasing expo-
nentially. New network applications—from Internet telephony to peer-to-peer file sharing—continue to appear. And new security threats, in the form of worms and viruses, and techniques for denial of service, are constantly appearing.

What is the internal behavior of the Internet and how do you infer it? How do you model its behavior? And how can the Internet be controlled in an orderly manner? These challenges are at the core of Distinguished Professor Donald Towsley’s research. One of his primary aims is to develop a framework within which network measurements can be designed and understood.

Theoretically, it’s possible to determine the performance of routers and communication links through simple queries. In practice, this would require administrative access to these devices. Unfortunately, any one organization has access to only a small fraction of the network’s internal nodes and commercial factors often prevent organizations from sharing internal performance data. End-to-end measurements, on the other hand, are easily executed.

Don, with his colleagues in the nearby Department of Mathematics and Statistics and at AT&T Labs-Research, has pioneered the field of network tomography. “It’s very similar to CAT scans in medical diagnostics,” he says. A CAT scan consists of a sequence of x-rays of the body taken from many different angles. Called slices, these x-rays are then correlated using sophisticated statistical techniques to form a three-dimensional view of the body.

Don’s team has developed a similar approach based on transmitting a sequence of small packets, or ‘probes’, from a sender to a set of receivers over a so-called multicast distribution tree. The results of the probes (loss or delay) correlate to produce estimates of link-level performance—packet loss rates and delay statistics of the links comprising the tree. The view of the network taken from the perspective of one sender and its collection of receivers resembles a slice.

“We then perform the same step for many different senders and collections of receivers and, ultimately, correlate the results of all these slices to form a detailed view of the interior of the network. This approach is incredibly useful for pinpointing problem spots on the Internet,” he explains.

The first approach, using multicast, Multicast Inference of Network Characteristics (MINC), was developed under the DARPA Next Generation Internet program. Since multicast service isn’t always available, Don and his team have extended the approach to work without it, by sending tightly grouped packets from a sender to two or more receivers. Each group now behaves very much like a single multicast probe sent to the same set of receivers. This approach lacks the resolution of MINC but is able to identify link-level loss rates as low as 0.5 percent.

Now Don is shifting his attention to developing a theoretical foundation for internal network measurements. The goal is to create an information theoretic framework suitable for designing low-cost, low-overhead network measurement infrastructures.
Collaborative web
The graph at right shows connections among our faculty, which indicate a high degree of collaboration and cross-discipline research. Red lines indicate co-authorship, orange lines indicate co-PIs on grant funding, and yellow indicates synthesis collaborations. The synthesis program involves graduate research projects between two faculty in distinct fields.
Stream processing for fast, scalable delivery

In an era of information abundance and ubiquitous networking, emerging applications require large-scale aggregation of distributed information sources, such as news sites, blogs, receptor devices, and network monitors, for timely delivery of high-value content. Key requirements of these systems are the ability to specify diverse customer information interests, efficient generation of the desired information from continuously arriving data, and scalability across multiple sources for numerous consumers.

To support this emerging class of distributed information systems, Assistant Professor Yanlei Diao is developing flexible and adaptive data management middleware for streaming data. Under a publish/subscribe interaction model, XML (eXtensible Markup Language) formatted content is delivered to recipients based on their information interests specified in the subscription. A key feature of her research is the use of continuous queries to represent such interests and efficient execution of these queries over XML streams.

Processing real-time XML streams raises considerable computational challenges compared to conventional relational database queries. Yanlei has developed the YFilter/ONYX system that employs a combination of novel XML processing techniques and traditional relational techniques for efficient query processing. To achieve scalability, the system further exploits commonalities among multiple queries for sharing in computation.

One important, new application is the real-time processing of data collected from networks of sensor devices, such as radio frequency identification (RFID) tags. For example, in supply chain management, RFID-tagged items are continuously monitored as they move from warehouse to transport to shelf to checkout.

In addition to tracking individual items, it would be helpful to analyze streams of RFID events in real time to make inferences about trends and to detect anomalies like shoplifting. Such stream-based processing, involving filtering, recursive pattern matching, and aggregation, requires new tools for efficiently formulating and executing a unique class of complex event queries. For this purpose, Yanlei has developed SASE, event processing middleware that facilitates the expression and execution of such queries over real-time RFID streams.
Literature reveals network of connections

If publishing is the lifeblood of academic research, accessing the literature is the heartbeat. Not surprisingly, Computer Science has been ahead of many academic disciplines in providing electronic access to published manuscripts. These papers from journals, proceedings, and technical reports are widely accessible through customized search engines such as CiteSeer, Google Scholar, and the ACM Portal. However, now Associate Professor Andrew McCallum is moving beyond cataloging papers with a new application for systematically collecting, parsing, and mining research papers.

The system, called Rexa, crawls the web for manuscripts and applies state-of-the-art probabilistic information extraction (IE) to confidently identify the attributes of a manuscript (such as the author names, title, abstract, venue, institutions and citations). Then it resolves these attribute strings to create de-duplicated, cross-referenced “first-class objects,” including people, papers, grants, universities, conferences, journals, research groups, and topics. The resulting inter-connected web of entities has provided new insights into the influence of manuscripts, and how ideas travel through social networks of researchers. Clustering methods are being applied to texts to automatically create topic hierarchies and specialized lexicons. Using Rexa, peer reviewers will soon be able to retrieve recommendations for related papers to assist in manuscript reviews, and program chairs will be able to obtain lists of recommended reviewers for papers.

The core scientific research in the Rexa project is to tightly integrate natural language processing, IE and data mining—enabling new methods for analysis and decision-making from distributed unstructured text. This work, in turn, pushes forward McCallum’s fundamental research on inference and learning in graphical models as well as on the integration of first-order logic with probability.

Rexa, which makes use of more than 100 compute servers to crunch through over 7 million research papers and 1 million de-duplicated authors, can be found online at http://rexa.info/.

Andrew’s research is only part of the large research effort in Information Retrieval. The 15-year old Center for Intelligent Information Retrieval (CIIR), directed by Distinguished Professor Bruce Croft and Associate Professor James Allan in collaboration with Andrew and Research Associate Professor R. Manmatha, is one of the leading IR research labs in the world. Bruce and James work together on a number of projects dealing with unstructured information—mostly text—and have developed new models of queries, documents, and relevance that are widely used by other researchers. Jointly with Carnegie Mellon University, they developed the Lemur toolkit and the Indri search engine, both popular open source projects in language modeling and information retrieval research. Indri is designed to support large scale IR, supporting both massive collections and high volume querying.

One new area of research is the application of IR concepts to statistical machine translation into English from other languages. By forming a topic model, erroneous translations can be discarded because the language use is inconsistent with the topic—e.g., a news story on coastal flooding is unlikely to use words associated with medical advances. A similar technique can be applied to speech recognition as well as machine translation where the approximate matching techniques at the core of IR are used to generate better estimates of the likelihood of phrases—providing more accurate recognition with less training data.
Buses and turtles have a lot in common, and it is not just their sometimes crawling pace. At UMass Amherst, Computer Science faculty Brian Levine, Mark Corner, and Emery Berger are building sophisticated networked systems inside buses and on the shells of turtles.

Keeping tabs on buses and turtles

Buses and turtles have a lot in common, and it is not just their sometimes crawling pace. At UMass Amherst, Computer Science faculty Brian Levine, Mark Corner, and Emery Berger are building sophisticated networked systems inside buses and on the shells of turtles.
The bus system, known as DieselNet, provides a useful service for bus passengers and is the largest known testbed for research in disruption-tolerant networks (DTNs). DTNs allow for routing in networks that are unstable due to mobility, low node density, short radio range, intermittent power, environmental interference, and even malicious intent. These sorts of problems commonly occur when an infrastructure is destroyed by natural disaster or war, but they also occur in wildlife monitoring, sensor networks, and underwater acoustic networks.

DieselNet, operational since May 2004, currently consists of 40 buses that travel around UMass Amherst and surrounding Hampshire County. Each vehicle powers a small Linux-based computer with an attached GPS and WiFi, which provides wireless connectivity to passengers, passersby, and other buses in the vicinity.

Custom software allows data to be transferred between buses as they pass on the road and to connect to open wireless access points in nearby buildings along their routes. In this way, buses convey their locations by exploiting multiple, unreliable, and indirect routes. For example, in the inset, red dots indicate bus-to-bus data transfers during a one month period. One of the key research goals is to develop communication protocols that are robust and efficient in such an unreliable and intermittent environment. Analysis of this real system has revealed characteristics that were not obvious in simulations alone.

Managing DTNs on buses may be complex, but networking a threatened species of turtle presents a number of additional (and unusual) challenges.

The Wood Turtle (*Clemmys insculpta*), found throughout the Northeast, lives along streams and in nearby woodlands. Its numbers are dwindling through loss of habitat and highway mortality, but conservation efforts to study the creatures in their natural habitat have been hindered by a lack of tracking data. Researchers currently track turtles manually using radio telemetry and are practically limited to recording a location every two to three days per animal. In order to more accurately understand how these turtles behave and use their habitat, Mark (pictured center with graduate students Jacob Sorber, left, and Michael Jones) is developing a new tracking system to collect more frequent and detailed data.

Engineering a sensor suitable for the back of a turtle highlights many of the challenges of a DTN device. The mounted sensor contains a GPS receiver and processor and must be small and lightweight. Power consumption is a key concern, since a typical GPS receiver will completely drain a small battery in two hours. Luckily, the cold-blooded turtles need to sun themselves, so a small solar panel can recharge the battery. Still, programming the devices requires careful consideration of fluctuating energy availability and demand.

Emery and Mark have designed a special-purpose high-level programming language called eFlux for writing energy-aware applications like those that will be mounted on the turtles. eFlux builds on Flux, a data-flow coordination language that they originally developed for high-performance servers. This language simplifies the programming of writing applications for network-embedded sensors, which typically consist of a sequence of tasks performed in response to events such as an expired timer or the arrival of a network message.

eFlux supports an application adaptation policy that includes alternate control paths and adjustable timers. Thus, as power needs and availability vary, the underlying runtime system can automatically adjust to a new environment without modifying the application. Also, the data-flow paradigm enables detailed program profiling and performance prediction.

Besides tracking buses or turtles, Brian, Mark, and Emery’s work has generated important insights into the deployment of low-power, robust sensor nodes in hostile environments, with broad applications to such domains as emergency management, remote sensing, and battlefield communication.
UMass Amherst computer scientists are bidding goodbye to the desktop, office, and machine room to venture onto the tornado-prone plains of Oklahoma, into the mountains of Puerto Rico, and above the rainforests of South America—thanks to continued miniaturization of sensing and computing devices.

This phenomenon is enabling a new generation of data-driven, networked sensing and actuating devices to be embedded into our physical environment. The result will be sensor networks that provide the ability to monitor our physical environment, detect and respond to significant events, and ultimately better understand the world in which we live.

The Center for Collaborative Adaptive Sensing of the Atmosphere (CASA) is a National Science Foundation Engineering Research Center that brings together a multidisciplinary group of engineers, computer scientists, meteorologists, sociologists, students, and industry and government scientists to conduct fundamental research, develop enabling technology, and deploy prototype systems based on a new paradigm of dense networks of highly-agile, low-powered sensors (radars) that sense the lowest few kilometers of the earth’s atmosphere.

Jim notes that CASA is “a wonderful illustration of how computing can impact and infuse many disciplines, which in turn provide us with a rich set of applications-driven, computing research challenges.”

The Wireless Sensor Networks Lab recently established by Deepak, Prashant and Mark Corner focuses on building scalable and energy-efficient sensor networks. Their research spans systems software and algorithms, networking protocols, embedded hardware, and sensor data processing algorithms. Lab projects include storage-centric sensor network.
architectures, low-power camera sensor networks, energy-efficient multi-processor, multi-radio and multi-storage sensor platforms, and application of their research to structural and transportation engineering.

The goal of the Aerial Imaging and Remote Sensing Lab (AIRS), led by Dr. Howard Schultz, is to advance the environmental sciences through computer vision techniques. For example, the NSF sponsored Forestry Project, in collaboration with Professors Allen Hanson, Jim Clark from Duke University, and Thomas Millette from Mount Holyoke College is developing methods for studying how forests are affected by and influence global weather. The team is building a highly automated aerial imaging system that enables biologists to obtain high-quality digital maps of environmental attributes with sub-meter geo-registration and elevation accuracies.

In another collaborative project with the Computer Vision Lab, software is being developed for creating 3-D models of cities from satellite images.

The image (right) shows a digital elevation model of central Boston in which heights are displayed using color gradients from red (below sea level) to yellow to green to blue (tall buildings).
The long evolution of data management technology has resulted in an enormous growth in data collection and storage. Electronic traces now exist for many of our activities and interactions: the things we read, watch, and buy, as well as the places we go, are very often recorded and stored by others.

Data collection on such a massive scale, if managed improperly, may violate our privacy if inappropriately released or modified, and allow our identities to be misused. Assistant Professor Gerome Miklau, who joined the faculty in 2005, is addressing these problems by developing new concepts for understanding the essential tradeoff between useful exchange of information and the violation of privacy.

The owner of sensitive data is faced with the challenge of permitting its legitimate use while protecting it from unauthorized disclosure. Data stored in a database can often be “viewed” in a way that is useful to the end user while omitting confidential data items. But Gerome recently showed how confidential data can be inferred, even when omitted from a view, based on associations among attributes.

His solution is a novel information-theoretic standard for determining when it is safe to publish a view without disclosing information about a sensitive query. When a view and a privileged query are secure, the view contains no information about the answer to the query, and the user will have no advantage in computing or guessing the answer. This standard for security offers data publishers a more accurate understanding of the disclosures they permit.

Another approach to managing data access is through symmetric key encryption—separate keys for each recipient. But in many scenarios, including database or general block-level file system access, a publisher may distribute content to many different receivers, often delivered through untrustworthy intermediaries, such as network proxies. Furthermore, over time the administrator will want to update or revoke keys. In such scenarios, the network overhead of key management of many keys becomes significant.

Assistant Professor Kevin Fu (pictured front with graduate student Tom Heydt-Benjamin), who also joined the department in 2005, has invented a “key regression” system that allows a publisher of encrypted content to efficiently distribute decryption keys to many readers and manage keys over time even when publishers have constrained bandwidth. Implementation is through a file system he’s called “Chefs.”

Although as much as 20 percent of digital data is believed to be stored in conventional database systems, a lot of sensitive data is found in non-conventional mobile devices. Kevin’s security research also includes the study of radio frequency identification (RFID) tags and similar devices.

Originally designed as a bar code replacement, RFID tags have become ubiquitous devices embedded in many everyday objects such as car keys, credit cards, inventory control, and passports. Yet the control mechanisms remain secret and often insecure. Kevin is building an RFID toolkit that provides emulation of RFID tags and receivers, allowing reverse engineering of RFID protocols and the development of secure RFID systems.

In one example (left), Kevin is considering the application of RFID to public transportation systems. With “smart” RFID bus passes, the system can offer more sophisticated fare structures. But the challenge is to ensure privacy of a passenger’s personal information encoded on the fare cards.
Spotting the face in the crowd

Since the mid-1980s, artificial intelligence researchers have made great strides in developing computer algorithms that can learn from examples. The trouble, says Assistant Professor Erik Learned-Miller, is that to accomplish this task computers need too many examples, whereas humans often grasp a concept in one go. “Currently,” he laments, “computers are just slow learners.”

Like many researchers in computer vision and machine learning, Erik has developed algorithms that can learn to recognize faces, read handwriting or identify diseases in medical images if given enough images of what these categories look like. For example, a typical program designed to learn handwritten letters from examples provides the computer with hundreds or thousands of examples of each letter of the alphabet. Designed to find patterns that characterize each letter, the program uses these learned patterns to recognize new letters even if they don’t exactly resemble any of the given examples.

Now Erik is trying to get computers to perform the same task when given only one image. People can do it, he points out, noting that when the symbol for the Euro, the new European currency, was introduced, people recognized it easily after seeing a single example, even if they later saw it in a different font, a different context or even in handwriting.

“My goal is to close the gap between human learning, which is highly efficient, and computer learning, which is abysmally inefficient,” he says. In his doctoral research, he made a great start—developing character classifiers that often achieved an enviable 90 percent accuracy even when trained on a single example of each character. His strategy was to “teach” the computer to draw on previously gained knowledge. “This seems to be the way humans work,” he explains.

Recently, working with Andras Ferencz of UCal-Berkeley and his own graduate students Vidit Jain and Marwan Matter, Erik has turned to more complex problems, such as recognition of faces or automobiles (see figure). Recognizing faces from one example is very difficult, he notes, since the look of a particular face depends on many factors—lighting, expression, viewing angle, facial hair, make-up, and so on. The key is to “teach” the computer what features are most salient and reliable over time and to use these features for recognition. But given the uniqueness of each person’s face, a generic concept of saliency may be unhelpful.

If computers could be made to learn effectively—and routinely—from one example, the impact would be dramatic. Erik envisions becoming partners with computers in learning about the world. It would be easy, he believes, to impart our knowledge to a computer or robot, instead of cumbersome as it is today. “It’s one of the most important problems in computer vision and machine learning research in the next ten years,” he says—and it’s one he’s intent on helping to solve.
Networks of intelligent autonomous agents

New applications composed of networks of intelligent agents that are hosted in open environments have attracted a lot of attention. These Multi-Agent Systems must adapt their behavior to the uncertain characteristics of their environment. Research in this area has had a long presence in the department and some of the seminal work has been done by two closely connected research groups: the Resource-Bounded Reasoning (RBR) group and the Multi-Agent Systems (MAS) group, headed by Professors Shlomo Zilberstein and Victor Lesser, respectively.

An example of a recent RBR project involved collaboration with NASA that aimed to make unmanned, planetary rovers more autonomous. As Shlomo explains, autonomously operated rovers on distant planets are faced with a great deal of uncertainty: “How long will operations take? How much power and data storage will be needed? What is the battery charging rate? Is the scientific task even possible?” Old practices were typically based on worst-case estimates. If an operation takes less time than expected, the rover waits for the next time-stamped operation. If operations take longer than expected, they may be terminated before completion. The new task planning algorithms developed by the group offer more efficient handling of the rover’s resources based on the current situation, thus minimizing delays and exploiting available resources.

Victor’s group has recently developed an adaptive real-time, distributed, sensor net- work for multi-vehicle tracking for the DARPA ANTs program. Large numbers of devices must communicate in real-time to allocate sensor resources and to schedule tracking responsibilities as vehicles enter and move through the environment. “It’s a complex distributed optimization problem that requires a changing organizational structure among devices along with real-time constraint-based optimization algorithms,” explains Victor. The approach taken was based on a new idea called “dynamic partial centralization.” The result was an efficient and effective tracking system that could scale to large numbers of sensor nodes and vehicles.

The ANTs project is one example of decentralized decision making. The computational complexity of such problems—a long-standing open problem—was fully characterized recently by Shlomo and Professor Neil Immerman, graduate student Daniel Bernstein, and postdoc Claudia Goldman. Together with Victor, these multi-agent researchers have formalized the problem in a new framework called decentralized Markov decision problems (MDP). Approximate and exact algorithms have recently been developed. “It really helped place multi-agent coordination on a more formal footing,” says Victor, one of the founders of the field, “and has led to much deeper understanding of the nature of coordination that has potential implications not only for other disciplines within computer science but also the broader scientific community.”
COMPUTER SCIENCE FACULTY

Micah Adler, Associate Professor
Ph.D., Computer Science, University of California at Berkeley, 1996
Interests: Algorithms, communication networks, parallel and distributed systems, and theoretical computer science.
Email: micah@cs.umass.edu

W. Richards Adrion, Professor
Ph.D., Electrical Engineering, University of Texas Austin, 1971
Interests: Multimedia learning technologies, impact of instructional technology on teaching and learning, analysis and verification of concurrent and real-time computing systems.
Email: adrion@cs.umass.edu

James Allan, Associate Professor
Ph.D., Computer Science, Cornell University, 1995
Interests: Information retrieval, event-based information organization, and minimally interactive retrieval and organization.
Email: allan@cs.umass.edu

George S. Avrunin, Adjunct Professor
Ph.D., Mathematics, University of Michigan, 1976
Interests: Concurrency, testing and analysis, requirements, real-time, and cohomology of finite groups.
Email: avrunin@cs.umass.edu

David A. Mix Barrington, Professor
Ph.D., Mathematics, Massachusetts Institute of Technology, 1986
Interests: Complexity of computation, circuit complexity, boolean function complexity, theory of automata, mathematical logic, and theory of algorithms.
Email: barring@cs.umass.edu

Andrew G. Barto, Professor
Ph.D., Computer Science, University of Michigan, 1975
Interests: Learning and planning in stochastic sequential decision problems; computational theory of motivation, reward, and addiction; models of learning and adaptation in animal motor control systems.
Email: barto@cs.umass.edu

Emery D. Berger, Assistant Professor
Ph.D., Computer Science, University of Texas Austin, 2002
Interests: Programming languages, memory management, garbage collection, operating systems, and runtime systems.
Email: emery@cs.umass.edu

Oliver Brock, Assistant Professor
Ph.D., Computer Science, Stanford University, 2000
Interests: Robotics, autonomous mobile manipulation, motion planning, computational biology, and computational geometry.
Email: oli@cs.umass.edu

Lori A. Clarke, Professor
Ph.D., Computer Science, University of Colorado, 1976
Interests: Software verification, testing, and analysis; software architecture and design.
Email: clarke@cs.umass.edu

Mark D. Corner, Assistant Professor
Ph.D., Electrical Engineering: Systems, University of Michigan, 2003
Interests: Mobile and pervasive computing, embedded systems, operating systems, file systems, and security.
Email: mcorner@cs.umass.edu

W. Bruce Croft, Distinguished Professor
Ph.D., Computer Science, University of Cambridge, England, 1979
Interests: Information retrieval and digital libraries.
Email: croft@cs.umass.edu

Yanlei Diao, Assistant Professor
Ph.D., Computer Science, University of California, Berkeley, 2005
Interests: Information architectures, database systems, data streams, data dissemination, XML query processing, and learning-based data processing.
Email: yanlei@cs.umass.edu

Kevin Fu, Assistant Professor
Ph.D., Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2005
Interests: Computer system security, secure storage, RFID security, web security, applied cryptography, and cryptanalysis.
Email: kevinfu@cs.umass.edu

Deepak Ganesan, Assistant Professor
Ph.D., Computer Science, University of California Los Angeles, 2004
Interests: Wireless sensor networks, distributed sensor data management, low-power networked embedded systems, pervasive computing, mobile computing, computer networks, and distributed systems.
Email: dganesan@cs.umass.edu
Weibo Gong, Adjunct Professor  
Ph.D., Applied Mathematics, Harvard University, 1987  
Interests: Complex modeling and control and communication security.  
Email: gong@cs.umass.edu

Robert M. Graham, Professor Emeritus  
MA, Mathematics, University of Michigan, 1957  
Interests: Operating Systems.  
Email: bob@cs.umass.edu

Roderic A. Grupen, Professor  
Ph.D., Computer Science, University of Utah, 1988  
Interests: Autonomous sensorimotor systems, embedded control, developmental dynamics, multifingered hands, walking machines, and animate vision.  
Email: grupen@cs.umass.edu

Allen R. Hanson, Professor  
Ph.D., Electrical Engineering, Cornell University, 1969  
Interests: Computer vision, image interpretation, mobile robotics, aerial image analysis, motion, stereo, 3-D reconstruction, sensor calibration, and education.  
Email: hanson@cs.umass.edu

Neil Immerman, Professor  
Ph.D., Mathematics, Cornell University, 1980  
Interests: Logic in computer science, computer-aided verification, complexity theory, and database theory.  
Email: immerman@cs.umass.edu

David D. Jensen, Associate Professor  
D.Sc., Engineering and Policy, Washington University, 1992  
Interests: Knowledge discovery and data mining, machine learning, and computing policy.  
Email: jensen@cs.umass.edu

Scott Kaplan, Adjunct Assistant Professor  
Ph.D., Computer Science, University of Texas Austin, 1999  
Interests: Virtual memory management and file system caching, reference trace collection and reduction, prefetching, and page replacement policies.  
Email: skaplan@cs.umass.edu

David C. Kulp, Assistant Professor  
Ph.D., Computer Engineering, Bioinformatics, University of California Santa Cruz, 2003  
Interests: Bioinformatics, sequence analysis, gene-finding, microarrays, variation and expression marker models, and data compression.  
Email: dkulp@cs.umass.edu

James F. Kurose, Distinguished Professor  
Ph.D., Computer Science, Columbia University, 1984  
Interests: Network protocols and architecture, network measurement, sensor networks, multimedia communication, and modeling and performance evaluation.  
Email: kurose@cs.umass.edu

Erik Learned-Miller, Assistant Professor  
Ph.D., Computer Science and Electrical Engineering, Massachusetts Institute of Technology, 2002  
Interests: Computer vision and machine learning, probabilistic and statistical methods in vision and image processing, non-parametric statistics, and information theoretic methods.  
Email: elm@cs.umass.edu

Wendy G. Lehnert, Professor  
Ph.D., Computer Science, Yale University, 1977  
Interests: Natural language processing and cognitive models of human thought processes.  
Email: lehnert@cs.umass.edu

Victor R. Lesser, Professor  
Ph.D., Computer Science, Stanford University, 1972  
Interests: Multi-agent systems, agents, real-time decision making, information gathering, situation assessment and interpretation of sensory data.  
Email: lesser@cs.umass.edu

Brian N. Levine, Associate Professor  
Ph.D., Computer Engineering, University of California Santa Cruz, 1999  
Interests: Information assurance, group communication, and mobility.  
Email: brian@cs.umass.edu

Sridhar Mahadevan, Associate Professor  
Ph.D., Computer Science, Rutgers University, 1990  
Interests: Artificial intelligence, cognitive science, machine learning, reinforcement learning, robot learning, and sequential decision making.  
Email: mahadeva@cs.umass.edu
R. Manmatha, Research Assistant Professor
Ph.D., Computer Science, University of Massachusetts Amherst, 1997
Interests: Multimedia indexing and retrieval, image and video retrieval, document image analysis, digital libraries, computer vision, and information retrieval.
Email: manmatha@cs.umass.edu

Andrew McCallum, Associate Professor
Ph.D., Computer Science, University of Rochester, 1995
Interests: Information extraction, knowledge discovery from text, statistical natural language processing, machine learning, and graphical models.
Email: mccallum@cs.umass.edu

Catherine McGeoch, Adjunct Professor
Ph.D., Computer Science, Carnegie Mellon University 1986
Interests: Algorithms and experimental algorithmics.
Email: ccm@cs.umass.edu

Lyle McGeoch, Adjunct Professor
Ph.D., Computer Science, Carnegie Mellon University, 1987
Interests: Heuristics for NP-hard problems, online scheduling algorithms, and dynamic graph algorithms.
Email: lam@cs.umass.edu

Kathryn S. McKinley, Adjunct Professor
Ph.D., Computer Science, Rice University, 1992
Interests: Compilers, memory management, runtime systems, and architectures with a focus on memory hierarchy performance.
Email: mckinley@cs.umass.edu

Gerome Miklau, Assistant Professor
Ph.D., Computer Science, University of Washington, 2005
Interests: Database research with an emphasis on security, database theory, and semi-structured data.
Email: miklau@cs.umass.edu

Robert N. Moll, Associate Professor
Ph.D., Mathematics, Massachusetts Institute of Technology, 1973
Interests: Knowledge-based systems, combinatorial optimization, and web-based education.
Email: moll@cs.umass.edu

J. Eliot B. Moss, Associate Professor
Ph.D., Computer Science, Massachusetts Institute of Technology, 1981
Interests: Object-oriented languages, performance measurement and optimization, garbage collection, and transactional memory.
Email: moss@cs.umass.edu

Leon J. Osterweil, Professor
Ph.D., Mathematics, University of Maryland, 1971
Interests: Process modeling and process programs, analysis of concurrency, software architectures, ecommerce and egovernment, medical processes and medical safety.
Email: ljo@cs.umass.edu

Krithivasan Ramamritham, Adjunct Professor
Ph.D., Computer Science, University of Utah, 1980
Interests: Real-time systems, transaction processing in database systems, and real-time database systems.
Email: krithi@cs.umass.edu

Edward M. Riseman, Professor Emeritus
Ph.D., Electrical Engineering, Cornell University, 1969
Interests: Computer vision, distributed sensor networks, mobile robotics, aerial image analysis, stereo processing, three-dimensional construction, and object recognition.
Email: riseman@cs.umass.edu

Edwina L. Rissland, Professor; NSF Program Director in the Information and Intelligent Systems (IIS) Division of the CISE Directorate
Ph.D., Mathematics, Massachusetts Institute of Technology, 1977
Interests: Case-based reasoning, AI and law, concept formation and evolution, CBR and information retrieval, and cyberlaw.
Email: rissland@cs.umass.edu

Sami Rollins, Adjunct Assistant Professor
Ph.D., Computer Science, University of California Santa Barbara, 2003
Interests: Distributed and peer-to-peer computing, wireless and mobile computing, Internet systems and applications.
Email: srollins@cs.umass.edu

Arnold L. Rosenberg, Distinguished Professor
Ph.D., Applied Mathematics, Harvard University, 1966
Interests: Theoretical and algorithmic aspects of modern computing environments, such as clusters and grids.
Email: rsnbg@cs.umass.edu
Prashant Shenoy, Associate Professor  
Ph.D., Computer Science, University of Texas Austin, 1998  
Interests: Operating systems, distributed systems, sensor networks, multimedia systems, and mobile computing.  
Email: shenoy@cs.umass.edu

Hava Siegelmann, Associate Professor  
Ph.D., Computer Science, Rutgers University, 1993  
Interests: Biological and physical computation, computational neuroscience, analog and adaptive information systems, and bioinformatics.  
Email: hava@cs.umass.edu

Ramesh K. Sitaraman, Associate Professor  
Ph.D., Computer Science, Princeton University, 1993  
Interests: Parallel and distributed systems networks, Internet services, algorithms, performance analysis, and theoretical computer science.  
Email: ramesh@cs.umass.edu

Ileana Streinu, Adjunct Professor  
Ph.D., Computer Science, Rutgers University, 1994  
Interests: Combinatorial and computational geometry.  
Email: streinu@cs.umass.edu

Donald F. Towsley, Distinguished Professor  
Ph.D., Computer Science, University of Texas Austin, 1975  
Interests: Networks and performance analysis.  
Email: towsley@cs.umass.edu

Paul E. Utgoff, Associate Professor  
Ph.D., Computer Science, Rutgers University, 1984  
Interests: Machine learning, representation induction, many-layered learning, learning from textbooks, feature construction, neural development, evaluation function learning, adversary search, and decision tree induction.  
Email: utgoff@cs.umass.edu

Arun Venkataramani, Assistant Professor  
Ph.D., Computer Science, University of Texas Austin, 2004  
Interests: Distributed systems and networking: operating systems, congestion control, fault-tolerance, security, and algorithmics.  
Email: arun@cs.umass.edu

William T. Verts, Assistant Professor  
Ph.D., Computer Science, University of Massachusetts Amherst, 1990  
Interests: Computer graphics and data structures.  
Email: verts@cs.umass.edu

Rui Wang, Assistant Professor  
Ph.D., Computer Science, University of Virginia, 2006  
Interests: Computer Graphics: global illumination algorithms, real-time rendering, graphics hardware, 3D scanning and image processing.  
Email: ruiwang@cs.umass.edu

Charles C. Weems, Associate Professor  
Ph.D., Computer Science, University of Massachusetts Amherst, 1984  
Interests: Computer architecture, interface between architecture and compilers and run-time systems, microarchitecture, memory systems, branch prediction, and parallel processing.  
Email: weems@cs.umass.edu

Jack C. Wileden, Professor  
Ph.D., Computer and Communication Sciences, University of Michigan, 1978  
Interests: Programming languages, inter-operability, databases, and bioinformatics.  
Email: jack@cs.umass.edu

Conrad A. Wogrin, Professor Emeritus  
Dr. Eng., Electrical Engineering, Yale University, 1955  
Interests: Computer vision and image understanding.  
Email: wogrin@cs.umass.edu

Beverly Park Woolf, Research Professor  
Ph.D., Computer Science, University of Massachusetts Amherst, 1984  
Ed.D., Education, University of Massachusetts Amherst, 1990  
Interests: Intelligent tutors, cognitive modeling, multimedia, and machine learning techniques to optimize tutoring.  
Email: bev@cs.umass.edu

Shlomo Zilberstein, Professor  
Ph.D., Computer Science, University of California Berkeley, 1993  
Interests: Artificial intelligence, heuristic search, planning and acting under uncertainty, decision theory, resource-bounded reasoning, models of bounded-rationality, multi-agent systems, and real-time decision making.  
Email: shlomo@cs.umass.edu
Our department has 72,000 square feet of spacious state-of-the-art offices and labs, over 900 computer systems running Unix, Mac, and Windows operating systems, high-performance computing clusters, an environmentally controlled server room, and a large full-time support staff dedicated exclusively to the department. The computers are connected via 100Mbs/1Gbs switched Ethernet and an 802.11 a/b/g wireless network, and connected to the campus via a high-speed link to the Internet and Internet2. Among our equipment is a UMass Humanoid Torso (Dexter), UMass Segway, UBot fleet of two-wheeled mobile robots, and a distributed sensor network that creates a “Smart Room” environment.
The town of Amherst offers UMass Amherst students all the features of a big city in the comfort of a historic, college-town setting.

It is located in the beautiful Pioneer Valley of western Massachusetts, just east of the Berkshire Hills, an area well known for its arts, cultural and historical attractions, and panoramic views. This central location and easy access to the major urban and capital cities of the Northeast makes the Amherst area popular for students and visitors. The campus is two hours from Boston and just three hours from New York City. Neighboring Northampton is a tourist hot spot with its wide array of shops and restaurants, and thriving cultural and arts scene.

As a member of the Five College consortium, the campus shares a mutually rewarding relationship with students from Amherst, Hampshire, Mount Holyoke, and Smith colleges.