You can subscribe to the CUCS news mailing list at
http://lists.cs.columbia.edu/mailman/listinfo/cucs-news/

CUCS colloquium information is available at
http://www.cs.columbia.edu/lectures

Visit the CUCS alumni portal at **http://alum.cs.columbia.edu**

Columbia University in the City of New York
Department of Computer Science
1214 Amsterdam Avenue
Mailcode: 0401
New York, NY 10027-7003

ADDRESS SERVICE REQUESTED

NON-PROFIT ORG.
U.S. POSTAGE
PAID
NEW YORK, NY
PERMIT 3593

CS@CU

NEWSLETTER OF THE DEPARTMENT OF COMPUTER SCIENCE AT COLUMBIA UNIVERSITY VOL.3 NO.2 SPRING 2007

From Computer Science to Fantasy:

An Interview with Naomi Novik

Imagine the year is 1806. You watch from a hilltop as Napoleon's army surges against Prussian forces that expect an easy defeat of the French Emperor. Suddenly, rising from the forest comes a flight of dragons, Aerial Corps in support of the French troops. The Prussian dragons and their crews of aviators are taken by surprise by the large numbers and unexpected tactics. The tide turns and the Prussians are forced to flee under heavy attack as the adventure continues.

This is the world imagined and realized by Naomi Novik in her Temeraire series of novels.

Born in New York in 1973, Ms. Novik studied English Literature at Brown University and Computer Science at Columbia University. After her studies at Columbia, she moved to Canada to help design and develop the computer game "Neverwinter Nights: Shadows of Undrentide". While working on the game, she found she preferred writing and returned to New York to write her first novel, "His Majesty's Dragon". This novel proved to be the first installment in her Temeraire series, a reimagining of the epic events of the Napoleonic Wars with an air force of dragons, manned by crews of aviators. The series has been wildly successful and was recently optioned by director Peter Jackson of "Lord of the Rings" fame.

CUCS Newsletter roving reporter Sean White recently had a chance to speak with Ms. Novik about her path to writing, her stories, and the similarities between designing software and novels.

Sean: You started in English and then moved to computer science and then into games. Can you tell me a bit about what led you to start writing?

Naomi: I started writing just for fun, just writing fan fiction in college – in undergrad, I mean – back when I was still an English major, pretty much when I first discovered the Internet. And then while I was still working on my English degree at Brown, at that point I started actually getting interested in computers

(continued on next page)

"once you get into writing a larger story it actually parallels very closely the process of what happens between writing a very short script to do a small specific task, and having to create the architecture for a large software project"

and working in the student support IT department that supported the student labs and really just enjoyed that work.

When I actually got a job after graduating I went to work for a company called D.E. Shaw & Co. I wanted to do a lot of technical stuff, so they let me do some work with the systems department despite the fact that I didn't have formal computer science training. And then after a while I decided that I wanted to go back and learn programming, and that I actually eventually wanted to work on computer games.

Computer games are one of those things that are a very nice combination of creative work, programming work, design, and creativity together, so I really liked the idea. Columbia has a great program called the Second Major program which basically lets you complete all the undergraduate requirements for a degree for a major that you didn't actually study in without having to take all the other electives and so forth. Going through the Second Major program allowed me then to apply for the master's program in computer science at Columbia. Then during the master's program I was funded as an MSTA, so I actually started doing some teaching. I taught one class about computer games and once I taught one on Windows programming, so that was a lot of fun.

And, basically, I then continued on and decided to go to the doctoral program after I finished the master's, mostly because I really enjoyed taking the classes and didn't quite want to stop. And it was a terrific experience. I've never particularly loved research so much; I've always preferred to actually build things in programming.

I got an opportunity to leave and work on the game *Neverwinter Nights: Shadows of Undrentide* after I had just finished my course work and was sort of flailing around, trying to figure out what I wanted to write my dissertation on. And then *Neverwinter Nights* came along and I said, "Okay. Never mind that dissertation," and took off to work on that. I really enjoyed working on *Neverwinter Nights*, but in fact the parts that I most enjoyed were the creative parts, the design, being able to craft the storylines and write dialogue for the characters. I'd been writing as a hobby for years and years by that point, but I pretty much wrote at the vignette level, not a larger story. And partly that's because once you get into writing a larger story it actually, I think, parallels very closely the process of what happens between, say, writing a very short script to do a small specific task, and having to create the architecture for a large software project.

Sean: Can you tell me more about that? That's interesting.

Naomi: Having worked in both fields, I absolutely believe that there's a parallel; once you're writing at the novel length you need a lot more structure, you need architecture that will support the writing. It's not enough to just be able to write decent prose, you have to also be able to put the events of the narrative together in a way that's going to be satisfying to the reader, and that will keep the pacing moving, will keep the events exciting to the reader as they go.

Working on *Shadows of Undrentide*, we were creating a 30 hour game and this plot had to engage the reader for that amount of time. So that was how we were approaching it; we wanted to create a general architecture for the story to hang off of. Doing that was very much a learning process for me in how you build the structure of the story of a narrative, and that's really what enabled me to make the jump in my own writing from writing at the short vignette level to writing a longer, more structured narrative.

I pretty much started thinking how I could still enjoy the creativity, the creative aspects, and a friend actually said to me "Why don't you sit down and try and write a novel?" And I said, "You know what? That's not a bad idea," and not too long after that started writing *Temeraire*. And *Temeraire* really just clicked for me right away and was just a joy to work on.

Sean: You describe the process as much like structuring software architecture. When you say that, I immediately think of black boxes to handle specific functions and a sort of hierarchical structure, or perhaps some object structure. Is that what you mean?

Naomi: The nice thing about writing a novel is you don't have to be as precise. But I do believe in some degree of compartmentalizing. For instance, my books, the way I write them is I don't see the book entirely as a single through line. I do view each chapter as doing something, each scene is doing something that then proceeds on to the next piece in very much the same way as classes or objects or functions within a program, expect of course its linear structure in general. But even though this is a series, the continuing adventures of Laurence and Temeraire, I try to keep each novel fairly self-contained as to plot, for instance, because I feel that's more satisfying to the reader than ending with a cliff hanger where the main plot of the story is unresolved and you're forced to wait for the next book to get resolution on the main plot thread. And, of course, there's the overall arching narrative of the Napoleonic Wars overlaid upon the entire series, which is the engine that kind of drives the whole process.

Sean: I was curious about that. Why the Napoleonic Wars?

Naomi: I love the Napoleonic era, the Regency era in England, it's just a lot of fun and has a nice combination of swashbuckling but at the same time a certain degree of modernity. The society is more etiquette driven, so it's later than pirates but before the Victorians. And there's a combination of the danger of the war, the anxieties produced by the war, and the Jane Austen comedy of manners that you have, these small domestic dramas that go on within society. So it's just a wonderful period to play in.

From a science fiction point of view it's a great era in which to put dragons because it's an era in which dragons are still enormously powerful. Once you get into an era like World War I or World War II where you have fighter planes available, then you have to make dragons a lot more powerful in certain ways, or you have do a lot more exposition and set up how dragons will work with the technology in order to still have dragons be powerful. Whereas in the Napoleonic era you can clearly see these people are sailing around in wooden sailing ships. It's very intuitive to immediately see why a fire breathing dragon is enormously terrifying and enormously powerful in this era, even while the fire breathing dragon can only fly 30 miles an hour and is still a believable physical creature, in a way; and also an intuitive creature to people who are used to dragons, who had some impressions of dragons from mythology and lore.

Sean: I take it you're also a big fan of historical fiction?

Naomi: I am. I love historical fiction. Patrick O'Brian's Aubrey/Maturin series is one of my absolute favorites and absolutely part of my inspiration, but Jane Austen is probably my single favorite author ever. The thing is, I find historical fiction and speculative fiction have a lot of the same pleasure, which is world building. Good speculative fiction for me creates a world that is unfamiliar and makes it believable to the reader and draws the reader in and creates characters to fit into that world and illuminate that world.

And historical fiction does the same thing because even though it's something that did exist at one time, it's still unfamiliar to us. It's still kind of an alien world to us. Most of us would feel as disoriented if we were dropped down in the middle of Jane Austen's regency England as we would if we were dropped onto Mars 100 years from now, and in some ways we would probably find it harder to adjust to going backwards and losing a lot of the creature comforts that we're used to, rather than adjusting to moving forward, I suspect.

Sean: Does that mean that you build a world first and then you think of your plot, or is it something that happens in parallel?

Naomi: What happens in parallel for me is world and characters. In the case of the *Temeraire* books it was very much the world, the historical world, that came first; I first thought of the idea of these characters and the Napoleonic wars, and then imagined "what if I had a dragon?" That's sort of the seed of it. And then basically from there what develops is what kind of a character would it be interesting to tell a story about in a world where the Napoleonic wars are going on, it's a fairly familiar regency England setting, but there are dragons.

That's how the character of Laurence developed. And Laurence starts out as a sea captain, a very successful man of his day, a gentleman and an officer who's doing very well and is very happy with his life. And then what happens is, of course, the advent of a dragon into his life throws his whole world out of kilter, which is much more interesting and exciting, I think, to read about a character who's a fish out of water. And that allowed me to take this character and introduce the elements that make the Temeraire universe different by having him be somebody who did not grow up around dragons, who's not used to dragons, he's learning about dragons even as the reader is. So those two aspects, the universe and the characters of Laurence and Temeraire were developed first. And then my plot invariably grows out of my characters.

Lois McMaster Bujold, who's a terrific writer and someone I admire a great deal, has said in interviews that what she likes to do is think of the worst possible thing that can happen to her characters and that's her plot, makes it happen and then forces her characters to deal with it.

And I think that's generally a good summary of the way that I approach plot. Because I am in a historical setting I have some constraint, and I like having some constraint in terms of what actually happened in the Napoleonic wars. I am changing history to some extent as I go, but I tend to only let myself get away with changing it for a reason, for a particular development reason. Then what I've done is tried to constrain myself a little bit because I find that it actually makes the work richer when I do that, when I don't let myself get away with easy solutions.

So within the constraint of the historical record, my plot then basically comes out of what would be interesting to have happen to my characters, what would be difficult for them, what would be challenging for them, what would make them hurt or unhappy, and what lets you hit those emotional high notes that I think the reader really enjoys; at the same time making them satisfying and believable.

Sean: Let me ask you one last question. What drives you to write?

Naomi: Wow. You know what? I don't actually think I can give you an answer to that question. I grew up reading voraciously and telling stories for me is how I respond to other texts, to ideas. I enjoy writing fiction much more than I enjoy writing essays or papers. There's something much more fun for me in communicating ideas through fiction and through inventing characters than in trying to write essays, for instance. It's just more satisfying to me.

But I don't think there really is an answer than any writer can give you other than – well, you can hear me sort of stuttering and saying, "What do you mean? How could I not write?"

Message from the Chair

Henning Schulzrinne, Professor & Chair

Thank you for reading our spring 2007 edition of the Columbia Computer Science newsletter!

Before introducing some of the features in this edition, I'd like to speak to a topic that has generated a fair amount of discussion in our department, as I'm sure it has in others.

As appears to be the case elsewhere, CS undergraduate enrollments at Columbia University are stabilizing, albeit at a much lower level than during the boom years and probably settling at a lower level than the previous "bottom" in the mid-1980s. The drop-off in enrollments seems to be nearly universal across North America, Western Europe and Australia, even where outsourcing is far less a media buzzword. One particularly bad early warning indicator is that the fraction of incoming freshmen that list CS as their probable major has dropped from a high of nearly 5% in 1983 and a second peak of 3.7% in 2000 and 2001 to just above 1%, a value last seen in 1977. Reflecting this decreasing student interest, the number of newly declared CS majors has decreased from its Fall 2000 high of 16,000 to 8,000 in Fall 2005.

The enrollment decrease seems to be fairly uniform across four-year institutions, although programs emphasizing computing skills, including those for non-traditional students, appear to have fared a bit better. From all I can tell, Columbia CS majors, at all degree levels, seem to have little problem finding decent jobs, even in fields, such as software development, that supposedly have migrated to Bangalore years ago. Speaking from local experience, we have significant difficulties finding capable system administrators, junior or senior, for our departmental computer operations.

At Columbia, we have also celebrated our 25th anniversary not too long ago; this offers a good opportunity to reflect on how we intend to reach, educate and inspire the next generation of computer scientists and computer engineers.

Organizations such as the CRA have been thinking about how to use media campaigns to reach teenagers, to inspire and convince them to choose STEM careers, in particular computer science. This can be very valuable, but I believe we need to go beyond improving the image of computer science in the eyes of parents, guidance counselors and teenagers. If we want to do more than focus on a handful of inspiring anecdotes, it would be helpful to know more about our graduates. This knowledge is also necessary for discussions about what we want to teach our students and what we can afford to omit, given that we have more than filled the available credit hours with material covering multiple sub-disciplines that are now as large as computer science was as a whole 20 years ago.

We seem to know surprisingly little about our students and about the paths they take once they leave us. For incoming students, what led them to choose computer science? Did they have specific professional goals in mind? Is CS a step along the way, as a pre-med, pre-business or pre-law degree? Were they guided by a computer scientist? Did they have the good fortune of having an excellent high-school CS teacher or participating in a RoboCup? Or did they believe that all CS majors program computer games? Did the lack of out-of-the-box programmability of modern PCs and the sophistication of commercial software discourage tinkering and exploration?

Given that the number of freshmen interested in CS has dropped even further than the number of majors, it would be useful to know why. To reflect some of the common stereotypes: Are they perceiving the field to be less exciting than ten years ago? Are high-school seniors scared off by tales of working conditions only suitable for single men without a social life? Do they see the choice of CS as risky, where a job may be available upon graduation, but might disappear a few years later, before the college loan has been paid off? Do highly-motivated students see a higher financial upside in professional careers in law, business or medicine or more societal benefit and relevance in biomedical engineering?

I suspect that one factor that may be discouraging high-school seniors might actually be one of the strengths of a computer science and computer engineering education – the extremely wide range of industries and job descriptions that a graduate might find themselves in. We still tend to think of CS majors pursuing a career in shrink-wrapped software, and CE majors developing the next generation of micro-processors, although those career paths probably have become a rare exception. I suspect students have a much better notion of what, say, a graduate in financial engineering (a very popular new program at Columbia) will be doing, even if their anticipation of glamour and financial opportunity may be misplaced.

While we might have some knowledge of the companies that fresh BS and BA graduates start at, we seem to know almost nothing about their career paths five and ten years later. What fraction of our undergraduates eventually pursue a post-graduate degree, either in CS or another discipline? How long do graduates remain involved in deeply technical work, as opposed to, say, project management? How does the technical career duration compare to older science and engineering disciplines? Is it true that somebody laid off at age 40 will have to sell real estate? How happy are CS and CE majors at the mid-point of their careers compared to other engineering graduates? What are the most exciting and intellectually rewarding aspects of their jobs? What contributions do they seem themselves making to their employers and larger society? What technical subjects and knowledge have turned out to be the most useful, both as a ticket to the first job and as a solid foundation for a technical career? Would our students recommend computer science as a career to their sons, daughters, nieces and nephews?

As a relatively young discipline, this lack of knowledge was understandable twenty years ago, but now our first graduates are reaching AARP eligibility, and Columbia is hardly the oldest CS department.

The CRA Taulbee survey is a good example of how CS has turned from anecdotes to data in evaluating and improving PhD programs. I believe we need to do the same for our undergraduate program, but focusing on outcomes, careers and motivations. Only by knowing what motivates our students, and those who have gone elsewhere, can we make effective changes in our programs and try to counteract stereotypes and myths.

Columbia Computer Science is in the process of exploring new majors and concentrations, such as those emphasizing information science and digital media, that better reflect the diversity of career paths and the breadth of the field. I would very much appreciate hearing from others that have explored these issues and may even have answers to some of the questions above.

The current edition of the newsletter reflects some of the diversity of computer science at Columbia. Beyond the use of CS as a pre-professional pathway, the interview with Naomi Novik illustrates how a degree in CS can indirectly lead to a career as a successful novelist. Prof. Steve Unger's article on e-voting emphasizes that knowing the limitations of our own tools is an important function of computer scientists (as opposed to the sales and marketing side of the industry...). Columbia has had a strong networking and distributed systems presence across both the Computer Science and Electrical Engineering departments, with the DNA lab featured in this issue as one important component of that research thrust.

As always, I look forward to hearing from you as our readers.

The **Distributed Network Analysis** (DNA) Group

The Distributed Network Analysis (DNA) Group is a group run jointly between the CS and EE departments at Columbia.

The group includes Professors Vishal Misra (CS and EE) and Dan Rubenstein (EE and CS) and nine PhD students, Eli Brosh (CS), Hoon Chang (CS), Hanhua Feng (CS), Kyung-Wook Hwang (EE), Abhinav Kamra (CS), Patrick Lee (CS), Richard Tianbai Ma (EE), Raj Kumar Rajendran (EE) and Joshua Reich (CS). The strong CS-EE collaboration should come as no surprise: Dan has a PhD in CS and is a faculty member in EE, while Vishal has a PhD in EE and is a faculty member in CS!

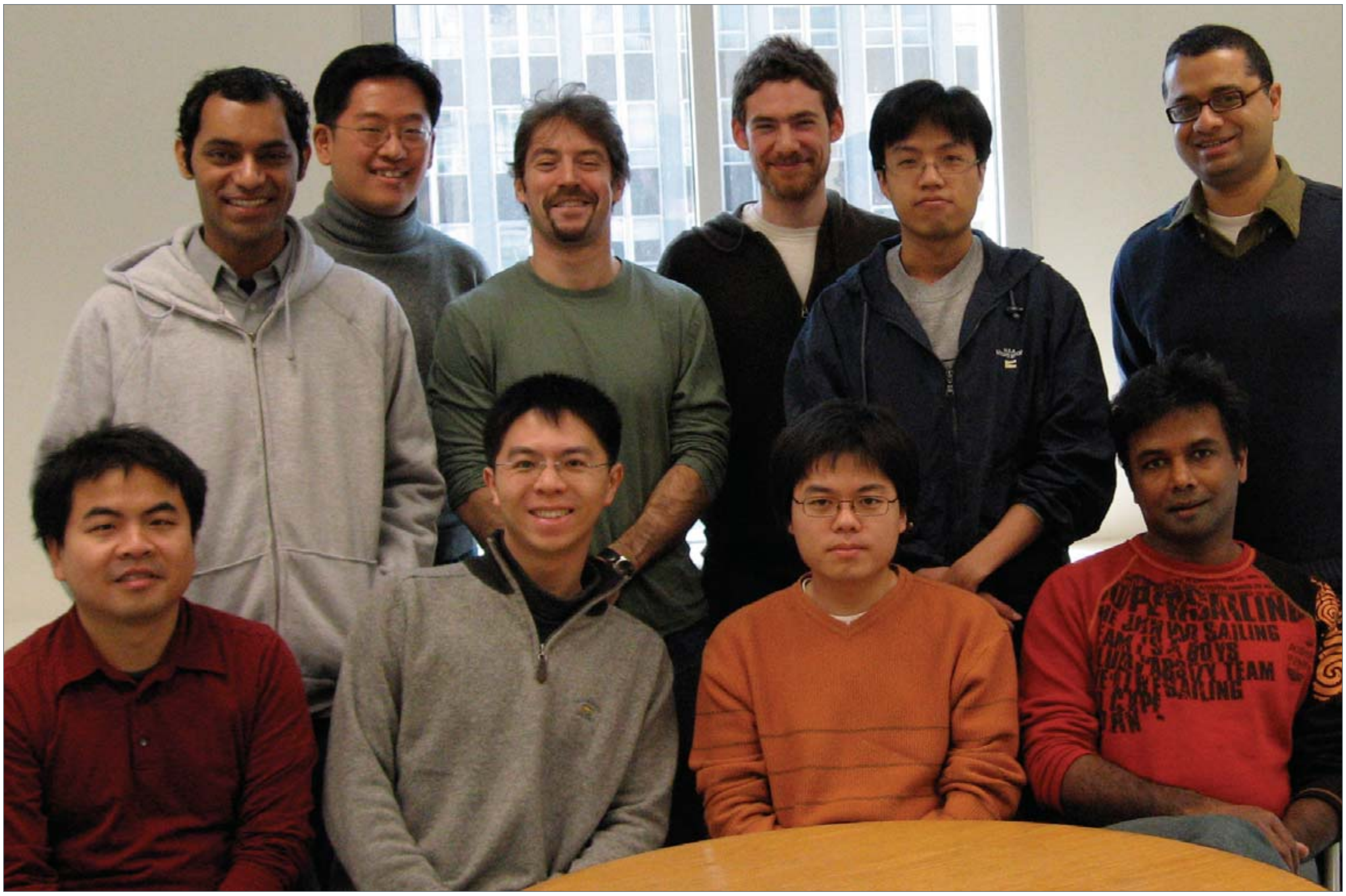
Our work involves analyzing and designing distributed systems, be it classical networking systems like wireless/sensor networks, peer-to-peer networks, or job processing systems like server farms. One of the main thrusts of our research has been on **resilience**. Our group's main focus is the use of a wide variety of analytical techniques (signal processing, graph theory, algorithms, control theory, queueing theory, optimization, and stochastic modeling) that enable us to explore networks scaled to ultra-large sizes under irregular conditions that are difficult to induce within simulation and experimentation. We also perform simulation and experimentation to provide additional support to our theoretical findings. A sampling of some of the projects being run in our lab is described next, to give a flavor of our style of research.

A Tunable Scheduling Policy

The performance of a system that simultaneously processes several tasks depends in large part on the scheduling policy used to partition processing resources among currently active jobs. There is an abundance of literature that proposes and evaluates so-called **blind** scheduling policies, which use only the arrival time and service obtained to-date of an active job to determine its processing priority with respect to other active jobs. Example polices include First Come First Serve (FIFO), Processor Sharing (PS), and Last Attained Service (LAS), and each of those policies are "optimal" for different classes of jobs. Since each policy has a unique implementation, system designers must decide in advance which policy best suits their needs. As an alternative, we have designed a **configurable** blind scheduler that contains a continuous, tunable parameter. By appropriately setting this parameter, the scheduling policy employed by the scheduler can exactly emulate or closely approximate the three above policies, as well as implement policies whose behavior is a hybrid of the above policies. We have implemented our scheduler into the Linux operating system, and demonstrated that (i) we can emulate the behavior of the existing, more complex scheduler with a single (hybrid) setting of the parameter, and (ii) the ability to easily vary the behavior of the scheduler is beneficial in environments where the job size distribution may change over time.

Resilience of Distributed Algorithms

In any setting that involves a network, many important computations are carried out in a distributed fashion. Examples are routing-algorithms, parameter-estimation algorithms and distributed computations. These distributed algorithms assume cooperating, error-free behavior among the nodes; i.e. each node involved in the computation carries out its tasks faithfully. This assumption is valid when the nodes are homogeneous, are present in a benign environment, and are under central control. However if the environment is hostile allowing nodes to be compromised, or if nodes are heterogeneous allowing for different interpretation and implementation of the algorithm, or if there is no central control that can verify and ensure the validity



Members of the Distributed Network Analysis (DNA) group. Top row, from left to right: Abhinav Kamra, Hoon Chang, Dan Rubenstein, Josh Reich, Patrick Lee, Vishal Misra. Bottom row, from left to right: Hanhua Feng, Richard Ma, Kyung-Wook Hwang, Raj Kumar Rajendran. Not pictured: Eli Brosh.

of each node's execution of the distributed algorithm, the computation may fail. Whether a distributed computation will fail, the conditions under which it will fail, and the extent to which it will fail are important questions, answers to which will help us determine the robustness of the distributed computation. Until now research and analysis of distributed algorithms has focused on the convergence properties of distributed algorithms, while assuming that nodes in the computation function as they should. It is important and useful to analyze the properties of distributed algorithms under adverse conditions, where nodes may either malfunction or be malicious and do not faithfully follow the protocol of carrying out the computation.

In our work we are studying such resilience analysis. We focus on a particular aspect of resilience we call strong detection that grapples with the issue of whether and when nodes can sense that there is an error in the distributed computation. Toward such detection, we demonstrate results which show that, for a distributed protocol, certain classes of disruptive behaviors are detectable while other classes of disruptive behavior are not.

Growth Codes

In various communication networks such as sensor networks, distributed sensing sources each generate data and use one another as forwarding points to transmit information to sink nodes. This transmission has to take place in the presence of possible failures of nodes, which disrupts routing and leads to a loss of information. We have developed novel coding techniques, which we call Growth Codes, to expedite and increase the information that can reach the sink in the presence of such failures.

Growth Codes is a linear coding technique that encodes information via a completely distributed process. The technique not only ensures that the data sink is able to recover all the distributed data, but also that data is efficiently recovered when only a small number of codewords is received.

Growth Codes are novel in that the codewords used to encode a fixed set of data change over time. Codewords in the network start with degree one and grow over time as they travel through the network en route to the sink. This results in the the sink receiving codewords whose degree grows with time. Using such a design, if the network is damaged at any time and the sink is not able to receive any further information, it can still recover a substantial number of original symbols from the received codewords.

The application of Growth Codes extends beyond Sensor Networks, and in fact they are the first **distributed** channel codes developed anywhere. We are currently developing an application of Growth Codes to a bittorrent like file distribution application.

2006-7 **Distinguished** Lecture Series

The Computer Science Department enjoyed another successful Distinguished Lecture Series during the 2006-7 academic year. Organized by Professors Steve Feiner and Ravi Ramamoorthi, the yearly series brings critically acclaimed scientists and pioneers in academia and industry from across the country to Columbia's campus.

These distinguished speakers give talks about their research, field questions from the audience, and meet with faculty and students. The lectures are open to the public and announced and documented at our website *(http://www.cs.columbia.edu/lectures)*, and ensure that members of the department and the community at large see first-hand the latest and greatest computer science research emerging outside the Columbia campus. This year, six lecturers visited us and discussed a broad and exciting range of topics, which we summarize here.

These speakers helped make this year an exciting and inspirational one for computer science at Columbia. We are looking forward to next year, when we bring more of the greatest researchers to our department and hear about their work.

Computer Science and the Age of Services

Our first lecture was on September 11th by **Stuart Feldman** from IBM.

Dr. Feldman is Vice President, Computer Science, at IBM Research, President of the ACM, and the author of the original UNIX "make" utility. He addressed the economic importance of services, which has been rising significantly, in large part because of the exciting progress in recent years in our ability to model, implement, and use services computationally. Dr. Feldman covered the technological vision and practical challenges underlying services-oriented applications and infrastructures. He surveyed the research issues that need to be addressed, ranging from basic questions of algorithms, models, and distributed systems to very practical issues of deployment and industrial practice. Services models provide a higher level of abstraction, one requiring not just implementation and control but also purpose and management.

Biology as Computation

Leslie Valiant from Harvard visited on September 25th.

Professor Valiant is the T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics and the winner of the Knuth Award in 1997. He discussed how computational models play an essential role in uncovering the principles behind a variety of biological phenomena. His talk considered recent results relating to three questions: How can brains, given their known resource constraints such as the sparsity of connections and slow elements, do any significant information processing at all? How can evolution, in only a few billion years, evolve such complex mechanisms as it has? How can cognitive systems manipulate large amounts of such uncertain knowledge and get usefully reliable results? He showed that each of these problems can be formulated as a quantitative question for a computational model, and argued that solutions to these formulations provide some understanding of these biological phenomena.

Conceiving an Internet for Tomorrow: Design for Tussle

David Clark from MIT gave a lecture on October 18th.

Dr. Clark is Senior Research Scientist at the MIT Computer Science and Artificial Intelligence Laboratory, past chair of the Computer Science and Telecommunications Board of the National Academies, and the chief protocol architect in the development of the Internet from 1981-1989. His talk emphasized that the future of the Internet is not defined by new technology. Rather, he pointed out that the Internet of today is firmly embedded in a larger space of social, political and economic considerations, and it is the understanding and manipulation of these considerations that will allow progress toward a better network for tomorrow. For example, our current security problems are not just a technical failure, but a consequence of decisions we made about the balance of identity and anonymity for Internet users. He offered some specific requirements that an Internet ten years from now should meet, and some design principles for tomorrow: how to design a mechanism that tries to shape the larger context in which the Internet must function.

Virtual Cinematography: Postproduction Control of Viewpoint and Illumination

Paul Debevec, research associate professor at the University of Southern California, executive producer of graphics research at the USC Centers for Creative Technologies, and the first recipient of ACM SIGGRAPH's Significant New Researcher Award, visited on November 20th.

Professor Debevec's talk explained some of the technology behind modern filmmaking, in which actors filmed in the studio are placed into distant or imagined locations, and computer-generated creatures and characters are added in alongside the main cast. The central challenge in combining the real and the rendered is making it look like everything was shot at the same time with the same camera. Professor Debevec explained his work on developing novel ways to film actors' performances so that their viewpoint and their illumination can be crafted virtually as part of the postproduction process. He illustrated his talk with images from films such as *Spider-Man 2*, in which digital actors were created with devices his group developed.

Human Computing: The Rewards of Computer Science in Everyday Life

Elizabeth Mynatt of Georgia Tech visited us on March 7th.

Professor Mynatt is associate professor in the College of Computing and director of the Graphics, Visualization and Usability Center (GVU) at Georgia Tech. Her talk addressed how visions describing the pervasive use of computing technologies by the general population often fail to explain how these technologies made the transition from the research lab to the home. Prof. Mynatt provided a fascinating glimpse into the complex cycle of human assimilation, learning, and adaptation that is often required before a novel technology becomes an integral part of our everyday lives. She presented three case studies in which these processes are visible today: consumer medical technologies for chronic healthcare, gaming technologies and their potential path into the workplace, and home networking.

Reflections on the VLSI Design Revolution

Lynn Conway, one of the pioneers of VSLI design, was here on March 27 to give a talk that was also the Department of Electrical Engineering Armstrong Memorial Lecture.

Professor Conway is Professor Emerita at University of Michigan, the inventor of dynamic instruction scheduling, and a recipient of the Pender Award of the Moore School and the Wetherill Medal of the Franklin Institute. She provided a fascinating retrospective on the VLSI chip design revolution that swept through Silicon Valley in the early 80s. She explained the key factors that led to that revolution, including the complexities of existing chip design practices, the opportunities for simplification, the opportunities for performance improvement via scaling of chip dimensions, the opportunity for applying then-new personal computers as tools for chip design, and the opportunities for exploiting the then-new Internet for design collaborations and quick turnaround chip prototyping. Professor Conway then talked about where we are now and speculated about what the future may bring.

Further details of the lecture series are available online at: http://www.cs.columbia.edu/lectures

New Computer Science Courses

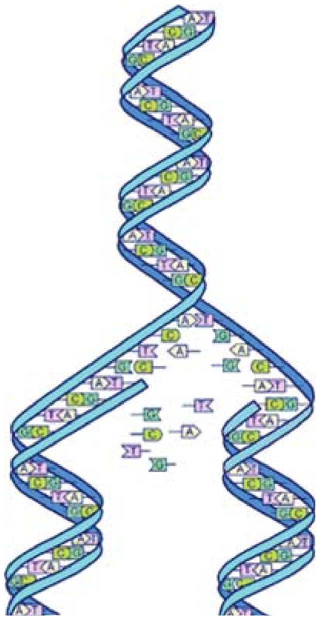
Life inside a computational camera.

Professor Peter Belhumeur offered COMS 6998-6, **Computational Photography**, in the Spring 2007 term. In recent years, the fields of computer graphics, computer vision and photography have converged to give rise to a new and very active area of research – Computational Photography. The goal in this work is to redefine the camera by using computational techniques to produce a new level of images and visual representations. The course is a seminar, offered to all students with knowledge in any of the three core areas: computer vision, computer graphics, or photography. Topics covered include HDR Imaging; Feature Matching; Image Mosaics, Image Stitching, and Dynamosaics; Image-Based Rendering, Environment Matting and Compositing; Image Refocusing; Motion Magnification; Removing Camera Shake; Camera Lens Arrays; Bluescreening; Programmable Lighting; Computational Flash Photography; Light Fields; Photo Pop-Up; Schematic Storyboarding; Face Detection; Face Modeling; Texture Synthesis; Video Textures; View Synthesis; Motion Estimation and Warping; Single and Multi-View Geometry; and Photo Tourism.

Discrete Differential Geometry is an emerging field with exciting applications in geometry processing and physical simulation.

Professor Eitan Grinspun taught COMS 4995-1, **Discrete Differential Geometry: An Applied Introduction** in the Spring 2007 semester. Differential geometry is the study of curves and surfaces. It gives us the language and structure to describe the behavior of physical systems (with applications to physical simulation) and to formalize the functionality of geometry-processing tools (with applications to remeshing, parameterization, smoothing, and compression). Researchers in a variety of areas have discovered that theories, which are discrete from the start, and have key geometric properties built into their discrete description, can often more readily yield robust numerical simulations and modeling tools which are true to the underlying continuous systems. By learning about such discrete theories we can create more realistic and robust physical simulations, and more intuitive and robust geometric modeling tools. This course introduces the student to the ideas of discrete differential geometry, geometric modeling, and physical simulation. The student learns about continuous and discrete differential geometric operators, and applies this knowledge to build their own mesh smoother (using discrete mean and Gaussian curvature), cloth- and thin-shell simulator (using a discrete Shape Operator), remesher (using a discrete Laplacian), and fluid flow visualizer (using Discrete Exterior Calculus and Hodge Decomposition).

Computer Science 6998-4, **Learning and Empirical Inference**, was offered in the Spring 2007 term. The course was taught by **Irina Rish** and **Gerry Tesauro** with lectures by **Tony Jebara** and **Vladimir Vapnik**. The course covers advanced topics for students interested in research in machine learning. These include Empirical Inference Science; Transduction; Universum; SVM+; Semidefinite Programming; Improved VC Bounds; Ellipsoid Machines; Fisher Discriminant; Metric Learning; Nearest Components Analysis; Xing Clustering with Side Information; Supervised Dimensionality Reduction; Metric Learning & Regression; Probabilistic connections to distances and Bregman divergences; Bayes point machines; E-Family PCA; Sparse optimization; Sparse PCA; Feature Selection; Matrix Factorization; l1-norm SVMs; Structured prediction; Max Margin Markov Nets; Conditional Random Fields; Reinforcement Learning; Active learning; Model Selection, Occam's Razor; BIC; AIC; Information Bottleneck Online Learning; Boosting; Bounds; Sampling; Invariance; and Groups.



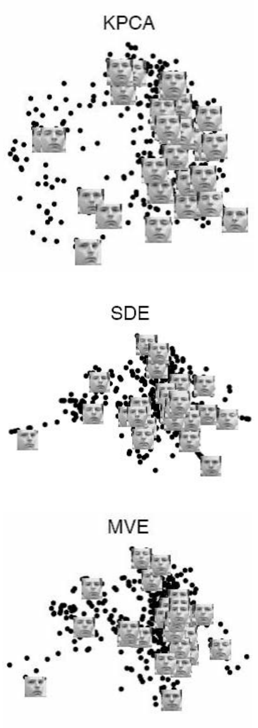
Professor Itsik Pe'er taught COMS 4995-2, **Seminar in Computational Genomics**, in the Spring 2007 semester. This course is intended to introduce students of both computational and bio-medical skill sets to current quantitative understanding of mammalian genomics and prepare them to computational research in the field. The course is interdisciplinary in nature, aiming at a broad scope of topics, including sequencing of the human genome, vertebrate genomes, highlighting parts of the genome, sequence conservation, sequence variation, structural mutations, primate evolution. The computational toolbox discussed includes parameter inference, likelihood analysis, hidden Markov and other graphical models, approximate string matching and other algorithms.

Visiting Professor Dragomir Radev

Professor Dragomir Radev (CUCS PhD '99, visiting from the University of Michigan) offered COMS 6998, **Search Engine Technology**, in the Spring 2007 term. A significant portion of the information that surrounds us is in textual format. A number of techniques for accessing such information exist, ranging from databases to natural language processing. Some of the most prestigious companies these days spend large amounts of money to build intelligent search engines that allow casual users to find what they want anytime, from anywhere, and in any language. This course covers the theory and practice behind the implementation of search engines, focusing on a wide range of topics including methods for text storage and retrieval, the structure of the Web as a graph, evaluation of systems, and user interfaces. A significant feature of the course is an open-ended research project in one of two categories:

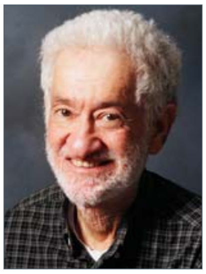
- Research paper – using the SIGIR format. Students are in charge of problem formulation, literature survey, hypothesis formulation, experimental design, implementation, and possibly submission to a conference like SIGIR or WWW.
- Software system – develop a working, useful system (including an API). Students are responsible for identifying a niche problem, implementing it and deploying it, either on the Web or as an open-source downloadable tool.

Professor Kenneth Ross taught COMS 6998-2, **High-performance Software for Modern Processors**, in the Spring 2007 semester. Modern processors have substantially different performance characteristics from those available even a decade ago. While traditional performance issues such as parallelism remain important, performance now depends much more on subtle architectural features such as cache behavior and branch misprediction effects. This course is about how to effectively utilize modern hardware when writing software. The emphasis is on getting high performance (usually measured as speed) for compute-intensive or data-intensive workloads. At the end of this course, students have learned about (and experienced) how to code algorithms so that they run efficiently on modern hardware platforms.



Three visualization algorithms showing a network of face images in a 2D visualization. Each image contains hundreds of pixels and dimensions but only two dimensions can be preserved. On the top, the plot just lays out the images according to two highest variability dimensions. In the middle, the network is stretched out before finding the top two dimensions. On the bottom (the most faithful visualization) the plot stretches the network of images in two dimensions while also squashing away the remaining unseen dimensions.

E-voting still vulnerable to fraud



Professor
Stephen Unger

The following piece by Professor Stephen Unger was printed in the **Journal News** *on October 26, 2006.*

A longer version is available at http://www1.cs.columbia.edu/~unger/articles/e-voting1-11-07.html

With the 2008 presidential election on the horizon, many computer engineers and scientists fear that electronic voting machines might facilitate voting fraud on an unprecedented scale.

Election fraud has always existed, regardless of the technology used. Both U.S. political parties have used such simple tactics as ballot box stuffing, ballot destruction, buying votes, and fraudulent counting, which, for conventional voting systems, are labor intensive and hard to conceal.

E-voting eliminates none of these "traditional" modes of cheating, while making possible entirely new corruption methods that are very hard to detect because detailed digital system operations are invisible to human observers.

E-voting systems have already crashed on election days. When there is Internet access, there are real dangers of hacker break-ins.

Still more serious is the possibility of systems with concealed features deliberately designed to falsify results by switching votes from one candidate to another. Engineers are accustomed to looking for program bugs and inadvertent design errors, and to testing systems for faulty circuit elements. Finding hidden features in a complex program, when even the comments may be deliberately misleading, is a very different, daunting problem, akin to detecting the tricks of stage magicians. Programs can be written that respond properly to all normally expected inputs during testing, but which change their behavior completely after some special input signal is entered.

Malware might be inserted during the election process by wireless transmissions, or under the guise of spelling corrections, or fixing "minor" program bugs. This code might become active only at certain times during the election process and then delete itself, leaving no trace behind.

Given the complexity of modern integrated circuit technology, with tens of millions of transistors on a chip, ensuring that the hardware side is clean would be at least as difficult.

Another corruption technique is a "denial of service attack." In precincts where the cheater's opponent is expected to receive a large majority of the votes, a substantial subset of the machines can be programmed to break down during the election, causing many voters to leave without casting ballots.

A proposed antidote to electronic manipulation is paper ballots. Voters, after verifying that the on-screen votes match their intentions, press a key causing the screen image to be printed on a paper ballot that they can see (but not touch). If the printed ballot is correct, they press another key, causing the ballot to be dropped into a slotted ballot box in full view of the voters and all observers. The system reports the tallies as computed from data corresponding to voter approved screen images or from optical scans of the paper ballots. Disputes over the validity of the final numbers can, in theory, be resolved by manual recount of the paper ballots, which are carefully preserved.

However, experiments indicate that most voters won't notice discrepancies between screen and paper ballots. Suppose a machine is programmed to change 10 percent of the votes cast for X to votes for Y both electronically and on the printed ballot, and that this is noticed by voters (and corrected) in 50 percent of the cases. Then 5 percent of X's votes would be recorded both electronically and on the paper ballots as votes for Y.

Post-election manual recounting of paper ballots and parallel testing (election day tests on randomly selected voting machines) have been proposed as safeguards. But election laws and the way they are administered in the various states offer little hope that these could generally be implemented reliably enough to thwart corrupt E-voting systems.

Every voting system is vulnerable to fraud if it is not open to public inspection or if representatives of concerned parties are not sufficiently vigilant at every stage. Ensuring that E-voting systems are not cheating may be impossible in the real world even with back-up paper ballots.

Having spent a lifetime working on the kind of technology underlying E-voting systems, I find myself in the peculiar position of advocating use of the most primitive type voting system: manually marked, manually counted, paper ballots. This simple, inexpensive technique has withstood the test of time, functioning reliably and scaling well for electorates of all sizes. It can be operated and monitored effectively by ordinary citizens, and is widely used (e.g., in New Hampshire, parts of several other states, and in most other countries, including Canada, France, Germany, and Sweden). E-voting systems confer no benefits justifying their great risks.

Department News & Awards

Adam Aviv, an undergraduate student in the Computer Science Department, was selected for Honorable Mention in the Computing Research Association's (CRA) Outstanding Undergraduate Award for 2007.

Professor **Peter Allen** is a member of a multi-institution team that has received a $5.3M NIH Biomedical Research Partnership grant for Cortical Control of a Dextrous Prosthetic Hand. The team includes researchers from Pitt, Minnesota, CMU, Arizona State and Columbia. The goal of this project is to build and demonstrate an anthropomorphic prosthetic arm and hand that is controlled by cortical output.

Professor **Steven Bellovin** received the 2007 National Computer Security Award of the National Institute of Standards and Technology and the National Security Agency. This prestigious honor, first awarded in 1988, recognizes individuals for scientific or technological breakthroughs, outstanding leadership, highly distinguished authorship or significant long-term contributions in the computer security field. The award was presented in a ceremony during the 22nd Annual Computer Security Applications Conference (ACSAC) in Miami Beach, Florida, in December 2006.

Professor **Luca Carloni** won an NSF Faculty Early Career Development (CAREER) award to develop a new communication-based design methodology for distributed embedded systems. The grant is titled "Integrating Control, Computation, and Communication– A Design Automation Flow for Distributed Embedded Systems".

Seung Geol Choi, a PhD student in the Department of Computer Science, won a best student-paper award at the International Workshop on Security (IWSEC'06). The paper was co-authored with Kunsoo Park (Seoul National University) and **Moti Yung** (RSA and Columbia University). The paper is titled "Short Traceable Signatures Based on Bilinear Pairings".

Professors **Steven Feiner** and **Kathleen McKeown** and their students are working in collaboration with Professor Desmond Jordan (Anesthesiology and Biomedical Informatics) and **Dr. Michelle Zhou** (IBM, PhD '99) on a charter project under the new IBM Open Collaborative Research program announced in December. They are developing tools to support physicians in creating, editing, and reviewing notes about patients. Notes are the primary source of information about patient status for caregivers, but are time-consuming to write and difficult to review quickly. To address this, the research emphasizes intelligent, ubiquitous information gathering; automated generation of notes; and on-demand synthesis of customized multimedia presentations of patient status. The team is using the New York Presbyterian Hospital Cardiothoracic Intensive Care Unit as a living laboratory in which to develop, demonstrate, and evaluate new technologies. The project leverages the researchers' collective expertise in human-computer interaction, natural language processing, computer graphics and visualization, cardiac anesthesiology, critical care medicine, and machine learning. Team members have been working together for over ten years building and testing experimental multimedia systems for caregivers in the Cardiothoracic Intensive Care Unit.

Professor **Steven Feiner** gave the keynote address at AUIC 2007 (the Australasian User Interface Conference) on directions for user interface research. He is also serving on the DARPA IXO Immersive Operations panel, which is exploring how effective systems could be created to communicate real-time data to personnel at all levels of command to enable quick and accurate decision-making. The goal of the panel is to suggest new program ideas for DARPA IXO to pursue. In addition, Professor Feiner was associate papers chair for ACM CHI 2007 and area chair for the 2006 IEEE and ACM International Symposium on Mixed and Augmented Reality, and was featured in a TechNewsWorld story on Augmented Reality (AR).

Agustin Gravano, **Christian Daniel Murphy**, **Joshua Reich**, **Cristian Petrus Soviani**, **Stanley Tzeng**, and **Ilho Ye** were named as "Distinguished TAs" for the Spring 2006 semester. Congratulations to our outstanding TAs!

Associate Research Scientist **Nizar Habash** of CCLS (the Center for Computational Learning Systems) was mentioned in a *Wired* piece on machine translation.

Professor **Tony Jebara** received an ONR award titled "Learning to Match Data from Heterogeneous Databases". The research proposal explores matching, b-matching and permutation within statistical learning, for applications including constraining social networks using graphs and b-matchings, visualizing large social networks, minimum volume embedding and merging social networks across heterogeneous databases. The applications will be explored via several novel algorithms and will scientifically advance the areas of b-matching, permutation, metric learning, structured prediction, invariance and graph embedding.

Professors **Angelos Keromytis** and **Sal Stolfo** won a Google Research Award to study "Safe Browsing Through Web-based Application Communities". Application Communities is a new paradigm for protecting software systems. Community members running independent instances of the same application will continuously exchange infor-

Department News & Awards (continued)

mation that allows them to collectively identify new faults and attacks (collaborative monitoring), and to automatically develop, test and apply fixes (heal).

PhD student **Christian D. Murphy** has garnered the highest TA honor in the School, the designation of "super TA." This designation is bestowed on TAs who have received the Extraordinary TA Award three times in the last four semesters (Fall '04–Spring '06). There were only three such super TAs designated in the whole School. Congratulations and thanks to Chris for his great teaching!

Professor **Steven Nowick** was awarded an ISE Grant (Initiatives in Science and Engineering) from the Office of the Executive Vice President for Research. The funding is for innovative proposals in early stages of development, with a special interest in cross-disciplinary work. The proposal is "Designing a Flexible High-Throughput Asynchronous Interconnect Fabric for Future Single-Chip Parallel Processors." The goal is to design a high-throughput, flexible and low-power digital fabric for future desktop parallel processors, e.g. those with 64+ processors per chip. This work is in collaboration with the parallel processing and CAD groups at the University of Maryland.

Professor **Rocco Servedio** was awarded a grant to participate in the DARPA Computer Science Study Panel. The objective of the Computer Science Study Panel is to rapidly identify ideas in the field of computer science that will provide revolutionary advances, rather than incremental benefit, to the Department of Defense. Areas of special interest include pattern recognition, computer vision, probabilistic reasoning, biologically inspired exploitation, abnormal behavior analysis, cognitive psychology, machine learning, and other advanced disciplines in computer science.

Professor **Joseph Traub** organized the 20th Anniversary Symposium of the Computer Science and Telecommunications Board. The symposium was the subject of substantial media coverage, including the New York Times.

Alumni News

Ely Labovitz has been promoted to chief technology officer for Accuro Healthcare Solutions, Inc. (Accuro), an industry leader in revenue management solutions. Ely holds a bachelor's degree in computer science from Yeshiva University in New York, and has completed master's courses in computer science from Columbia with concentrations in algorithm analysis, object oriented development, and computer architecture.

The shareholders of CHAMCO AUTO (China America Cooperative Automotive, Inc.) elected **William Pollack** as Chairman. William holds a BA in Mathematics and Computer Science and an MS in Computer Science from Columbia.

Naomi Novik, former CS department instructor and MS student of Professor Jason Nieh, was the subject of a New York Times Arts Section profile on the occasion of selling film rights of her three fantasy novels to Peter Jackson, the director of the "Lord of the Rings" movies. *(See cover article.)*

Daniel Yellin (PhD '87) has become Director of the IBM Software Group Lab in Israel. Daniel is an IBM Distinguished Engineer and has been working at IBM since graduating from Columbia.

Erez Zadok (PhD '01) was promoted to the position of Associate Professor with tenure at SUNY Stony Brook.

Hrvoje Benko

Advisor:
Steven Feiner

User Interaction in Hybrid Multi-Display Environments

Abstract: In recent years, the typical computer workspace has been experiencing a very significant transformation, from a single desktop computer with a single attached display, to a multi-display environment (MDE) with multiple connected devices. This dissertation presents the design, implementation, and evaluation of novel pointer-based and gesture-based interaction techniques for MDEs. These techniques transcend the constraints of a single display, allowing users to combine multiple displays and interaction devices in order to benefit from the advantages of each.

First, we introduce a set of Multi-Monitor Mouse techniques, which improve existing mouse pointer interaction in an MDE, by allowing users to instantaneously relocate (warp) the cursor to an adjacent display, instead of traversing the bezel. Formal evaluations show significant improvements in user performance when compared to standard mouse behavior.

Next, we focus on a particular kind of MDE, in which 3D head-worn augmented reality displays are combined with handheld and stationary displays to form a hybrid MDE. A key benefit of hybrid MDEs is that they integrate and utilize the 3D space in which the 2D displays are embedded, creating a seamless visualization environment. We describe the development of a complex hybrid MDE system, called Virtual Interaction Tool for Archaeology (VITA), which allows for collaborative off-site analysis of archaeological excavation data by distributing the presentation of data among several head-worn, handheld, projected tabletop, and large high-resolution displays.

Finally, inspired by the lack of an interaction vocabulary for hybrid MDEs, we designed three sets of gestural techniques that address the important issues of data transitions across 2D and 3D displays and precise interactions within those displays. First, we present Cross-Dimensional Gestures, which facilitate transitioning the data between devices, displays, and dimensionalities, by synchronizing the recognition of gestures between a 2D multi-touch-sensitive projected display and a tracked 3D finger-bend sensor glove. Second, we describe a set of Dual Finger Selection techniques that allow for precise and accurate selection of small targets within 2D displays, by exploiting the multi-touch capabilities of a tabletop surface. Third, we present the Balloon Selection technique, which was found to be three times more accurate than standard wand-based selection when selecting small 3D objects above a tabletop surface. Balloon Selection decouples the 3DOF selection task into a 2DOF task and a 1DOF task, while grounding the user's hands on a tabletop surface.

Sasha Blair-Goldensohn

Advisor:
Kathy McKeown

Long-Answer Question Answering and Rhetorical-Semantic Relations

Abstract: Over the past decade, Question Answering (QA) has generated considerable interest and participation in the fields of Natural Language Processing and Information Retrieval. Conferences such as TREC, CLEF and DUC have examined various aspects of the QA task in the academic community. In the commercial world, major search engines from Google, Microsoft and Yahoo have integrated basic QA capabilities into their core web search.

These efforts have focused largely on so-called "factoid" questions seeking a single fact, such as the birthdate of an individual or the capital city of a country. Yet in the past few years, there has been growing recognition of a broad class of "long-answer" questions which cannot be satisfactorily answered in this framework, such as those seeking a definition, explanation, or other descriptive information in response. In this thesis, we consider the problem of answering such questions, with particular focus on the contribution to be made by integrating rhetorical and semantic models.

We present DefScriber, a system for answering definitional ("What is X?"), biographical ("Who is X?") and other long-answer questions using a hybrid of goal- and data-driven methods. Our goal-driven, or top-down, approach is motivated by a set of "definitional predicates" which capture information types commonly useful in definitions; our data-driven, or bottom-up, approach uses dynamic analysis of input data to guide answer content. In several evaluations, we demonstrate that DefScriber outperforms competitive summarization techniques, and ranks among the top long-answer QA systems being developed by others.

Motivated by our experience with definitional predicates in DefScriber, we pursue a set of experiments which automatically acquire broad-coverage lexical models of "rhetorical-semantic relations" (RSRs) such as Cause and Contrast. Building on the framework of Marcu and Echihabi (2002), we implement techniques to improve the quality of these models using syntactic filtering and topic segmentation, and present evaluation results showing that these methods can improve the accuracy of relation classification.

Lastly, we implement two approaches for applying the knowledge in our RSR models to enhance the performance and scope of DefScriber. First, we integrate RSR models into the answer-building process in DefScriber, finding incremental improvements with respect to the content and ordering of responses. Second, we use our RSR models to help identify relevant answer material for an exploratory class of "relation-focused" questions which seek explanatory or comparative responses. We demonstrate that in the case of explanation questions, using RSRs can lead to significantly more relevant responses.

Hoon Chang

Advisors:
Vishal Misra and
Dan Rubinstein

Analytical Model and Fairness Scheduling of CSMA/CA in Physical Layer Capturing

Abstract: While physical layer capture has been observed in real implementations of 802.11 devices, there is a lack of accurate models that describe the behavior of the phenomenon and deep investigation for fair allocation. We first present a general analytical model and an iterative method that predicts error probabilities and throughputs of packet transmissions with multiple sender-receiver pairs. In our model, MAC protocol is considered a feedback controlling system and the least square method is used to model the rate adjustment of 802.11 DCF protocol. For the interaction and interference among nodes, we present an iterative method to offer a more accurate prediction than previous work by taking into account the cumulative strength of interference signals and using the BER model to convert a signal to interference and noise ratio value to a bit error probability. This permits the analysis of packet reception at any transmission rate with interference from neighbors at any set of locations. We prove that our iterative method converges and verify the accuracy of our model through simulations in Qualnet.

Second, we present log-utility fair scheduling in physical layer capturing. Having been known widely as a less egalitarian approach to max-min fairness, log-utility fair scheduling achieves a reasonable total throughput as well as acceptable fairness. We investigate the effect of log-utility fairness in physical capturing and present theorems regarding convex feasible allocation space in two-sender cases. Based on our analytical model, we also propose a decentralized log-utility fair allocation algorithm. The extension of our iteration method and analytical model yields an algorithm that returns access rates of all senders within a feasible running time, linear in the number of senders. For inputs to the algorithm, we present a protocol to gather neighbors' error statistics. We show log-utility fair allocations through simulations.

We also perform real experiments to verify our analytical model. With the laboratory trial network testbed, Orbit, we have developed measurement software for error probabilities and throughputs. The accuracy of our analytical model is verified with measured results in the field. Our fairness scheduling algorithm is implemented using PC and laptop nodes. To extend our algorithm, we consider quality-of-service constraints such as requirements of maximum acceptable error probability and minimum throughput. Assuming only the information of one-hop neighbors is available, we develop an extended distributed algorithm to meet given constraints. Another research topic is channel allocation. Considering channel allocation and access rate assignment together, we maximize the log utilities of nodes.

Michel Galley

Advisor:
Kathy McKeown

Pragmatic and Syntactic Models for Probabilistic Speech Summarizers

Automatic speech summarization can reduce the overhead of navigating through long streams of speech data by generating concise and readable textual summaries that capture the "aboutness" of speech recordings, such as meetings, talks, and lectures. In this thesis, we address the problem of summarizing meeting recordings, a task that faces many challenges not found with written texts and prepared speech. Informal style, speech errors, presence of many speakers, and apparent lack of coherent organization mean that the typical approaches used for text summarization must be extended for use with conversational speech. We illustrate how techniques that exploit automatically derived pragmatic and syntactic structures can help create summaries that are more on-target, coherent, and readable than those produced by current state-of-the-art approaches.

We present the discriminative learning of graphical models for summary sentence selection, evaluate different model structures, and discuss their impact on the coherence of generated summaries. Significant findings in this work show that dependencies between pragmatically related sentences, such as between a Question and its Answer, or an Offer and its Acceptance, are instrumental in determining which sentences should be included in the summary. These dependencies also help generating more coherent summaries, i.e., they penalize the inclusion of a Question if none of its Answers are included as well. We discuss and address some computational intricacies and inference problems specific to speech with many participants, which often incorporate pragmatic dependencies between sentences that are far apart in the speech stream.

We also investigate the problem of rendering errorful and disfluent meeting utterances into concise sentences that are maximally readable. We present a trainable syntax-directed sentence compression system that automatically removes information-poor phrases (such as "you know" or "for some reason") from the syntax tree of each input sentence. A novel aspect of this sentence compression approach lies in its Markovization of syntactic compression rules, a process that factorizes these rules into substructures more amenable for probabilistic estimation. Our decompositions not only facilitate supervised learning under severe sparseness conditions, but also enable the incorporation of many additional features, such as lexical dependencies and tree internal annotation, which are critical for accurately determining whether a given syntactic phrase is grammatically optional or mandatory. This sentence compression technique was effectively applied to both speech and written texts.

Edward Ishak

Advisor:
Steven Feiner

Content-Aware Interaction in User Interfaces

Whether interacting with a mobile phone or a wall-sized display, users often encounter screen space limitations that prevent the simultaneous display of all visual objects required to complete a task. Depending on the individual user and their current task, screen space limitations result from several factors, including insufficient physical space and inadequate pixel resolution. A number of techniques have been created to address these limitations, including overlapping semi-transparent windows, scrolling, and layout management. However, these techniques are typically designed to be generic and do not consider the properties of the content to which they are applied. In this thesis, we present an approach to making user interfaces content-aware by augmenting existing interaction techniques. By content-aware, we mean that they take into account various physical and semantic attributes of the content, such as size, location and type.

We designed, implemented, and evaluated content-aware versions of three existing user interface techniques: content-aware transparency, content-aware scrolling, and content-aware layout. Content-aware transparency applied to overlapping windows makes it possible for users to interact with otherwise hidden content by rendering important regions of windows opaque, and unimportant regions transparent, thus keeping overlaid contents legible and distinguishable at all times. Furthermore, based on properties of the overlapping material, appropriate image-processing filters are applied to obstructed content to help disambiguate the overlapping content. Content-aware scrolling allows a user to scroll along a system- or user-defined path within a document using conventional scrolling interactions, varying scrolling speed depending on content location. Content-aware layout lays out windows containing content relevant to that of the currently focused screen area, positioned relative to that area. We present quantitative user performance data gathered from formal experiments, as well as qualitative questionnaire feedback to show that interaction with content-aware techniques can provide an effective advantage over techniques that are not content-aware.

Risi Kondor

Advisor:
Tony Jebara

Group theoretical methods in machine learning

Abstract: If mathematics is about capturing structure in the world around us, then arguably there is no greater success story in the entire subject than the theory of groups and their representations. Having started out as a device for manipulating the roots of polynomial equations, groups have proved to be so fundamental to understanding the quantum world that to modern physics they are as indispensable as the real numbers.

Learning problems do not usually involve quantum particles of course, but they do confront non-trivial symmetries and harmonic analysis on various discrete and continuous spaces. In this regard it is surprising that up to now the connections to group theory, and algebra in general, have so far remained largely unexploited. Beyond painting a unifying theoretical picture, algebra also has algorithms to offer: recent years in particular have seen the development of a number of new efficient algorithms.

This thesis explores the connections between group theory and learning from different angles, directed at different applications. One overarching theme however, is the extension of regularization theory to groups, allowing us to deploy SVMs and other familiar statistical learning algorithms in this algebraic domain. A number of canonical theorems regarding positive definite functions on groups and their interpretations are provided.

The theoretical results find application in multi-object tracking, where a time varying distribution over permutations has to be maintained. Representation theory not only tells us how to evolve this distribution, but also tells us how to represent it compactly, and the modern theory of non-commutative FFTs provides algorithms for efficient updates.

Another broad area for applications is capturing symmetries, such as the rotation and translation invariance of natural images. I describe a fundamentally new approach to generating rotation and translation invariant features based on combining ideas from representation theory with some classical tools from signal processing.

Finally, I investigate how the structure of the symmetric group is reflected in ranking problems and their many variations, and how this can be exploited to build more principled and more powerful ranking systems.

Janak Parekh

Advisor:
Gail Kaiser

Privacy-Preserving Distributed Event Correlation

Abstract: Event correlation is a widely-used data processing methodology for a broad variety of applications, and is especially useful in the context of distributed monitoring for software faults and vulnerabilities. However, most existing solutions have typically been focused on "intra-organizational" correlation; organizations typically employ privacy policies that prohibit the exchange of information outside of the organization. At the same time, the promise of "inter-organizational" correlation is significant given the broad availability of Internet-scale communications, and its potential role in both software fault maintenance and software vulnerability detection.

In this thesis, I present a framework for reconciling these opposing forces via the use of privacy preservation integrated into the event processing framework. I introduce the notion of event corroboration, a reduced yet flexible form of correlation that enables collaborative verification, without revealing sensitive information. By accommodating privacy policies, we enable the corroboration of data across different organizations without actually releasing sensitive information. The framework supports both source anonymity and data privacy, yet allows for temporal corroboration of a broad variety of data. The framework is designed as a lightweight collection of components to enable integration with existing COTS platforms and distributed systems. I also present an implementation of this framework: Worminator, a collaborative Intrusion Detection System, based on an earlier platform, XUES (XML Universal Event Service), an event processor used as part of a software monitoring platform called KX (Kinesthetics eXtreme). KX comprised a series of components, connected together with a publish-subscribe content-based routing event subsystem, for the autonomic software monitoring, reconfiguration, and repair of complex distributed systems. Sensors were installed in legacy systems; XUES' two modules then performed event processing on sensor data: information was collected and processed by the Event Packager, and correlated using the Event Distiller. While XUES itself was not privacy-preserving, it laid the groundwork for this thesis by supporting event typing, the use of publish-subscribe and extensibility support via pluggable event transformation modules. I also describe techniques by which corroboration and privacy preservation could optionally be "retrofitted" onto XUES without breaking the correlation applications and scenarios described.

Worminator is a ground-up rewrite of the XUES platform to fully support privacy-preserving event types and algorithms in the context of a Collaborative Intrusion Detection System (CIDS), whereby sensor alerts can be exchanged and corroborated without revealing sensitive information about a contributor's network, services, or even external sources, as required by privacy policies. Worminator also fully anonymizes source information, allowing contributors to decide their preferred level of information disclosure. Worminator is implemented as a monitoring framework on top of a collection of non-collaborative COTS and in-house IDS sensors, and demonstrably enables the detection of not only worms but also "broad and stealthy" scans; traditional single-network sensors either bury such scans in large volumes or miss them entirely. Worminator supports corroboration for packet and flow headers (metadata), packet content, and even aggregate models of network traffic using a variety of techniques.

The contributions of this thesis include the development of a cross-application-domain event processing framework with native privacy-preserving types, the use and validation of privacy-preserving corroboration, and the establishment of a practical deployed collaborative security system. The thesis also quantifies Worminator's effectiveness at attack detection, the overhead of privacy preservation and the effectiveness of our approach against adversaries, be they "honest-but-curious" or actively malicious.

Angelos Stavrou

Advisor: Angelos Keromytis

An Overlay Architecture for End-to-End Service Availability

Abstract: Perhaps one of the most compelling problems of the Internet today is the lack of an overarching approach to dealing with on-line service security and availability: there exist a lot of mechanisms but no "security architecture"– no set of rules for how these mechanisms should be combined to achieve overall good security. This thesis is aimed at introducing and analyzing mechanisms that boost the security, resilience and performance of network systems. These systems are composed of large numbers of untrusted and unreliable components that communicate using the existing network infrastructure.

Towards this goal, we propose and evaluate practical mechanisms that can protect a wide range of services while maintaining or even improving their performance characteristics. Our end goal is to provide a practical end-to-end framework that significantly improves service availability and connectivity without incurring a prohibitive deployment or performance cost. Ideally, our approach should be able to scale to millions of users and accommodate any applications' requirements including network latency and throughput. We show our progression towards our end goal by presenting the advantages and limits of the protection system we developed: PROOFS, WebSOS, MOVE, and Spread-Spectrum with Multi-path overlays.

Xiaotao Wu

Advisor: Henning Schulzrinne

Ubiquitous Programmable Internet Telephony End System Services

Abstract: In Internet telephony, endpoints usually have CPU and memory, so they are programmable and can perform services such as call forwarding, transfer, and screening. In contrast, the traditional Public-Switched Telephony Network (PSTN) assumes dumb endpoints. In addition, peer-to-peer (P2P) technologies introduce telecommunication networks that do not need proxy or application servers to make calls. In such P2P networks, many telecommunication services have to be performed on end systems, such as intelligent phones. The enhanced capabilities of end systems and the service requirements in P2P networks motivate the investigation of end system services in this thesis.

Performing services in end systems may result in many new communication services, make telecommunication services more distributed, and make telecommunication networks more robust and efficient overall. At the same time, telecommunication services may become more difficult to manage, thus requiring new techniques for creating and composing services. This dissertation presents my research on defining end system services, developing efficient and user-friendly tools for creating end system services, managing end system feature interactions, and integrating end system services with other Internet services, such as web, email, location-based services, and networked appliance control.

I first analyzed the difference between end system services and network services. The analysis showed that they differ in call models, targeted service creators, and call control actions. Because of these differences, I defined a new scripting language called the Language for End System Services (LESS) specifically for end system service creation. LESS is designed to allow comparatively inexperienced end users to create services. End users may create conflicting services over time, so I developed a method that is based on the action conflict tables I defined in LESS and a tree merging algorithm to handle feature interactions among LESS scripts. In addition, because end users are often not aware of available services and do not know how to create services, automatically generating their desired services can be a great help to them. Therefore, I built a service learning system that uses the Incremental Tree Induction (ITI) algorithm to automatically create LESS scripts based on users' communication histories. Furthermore, to evaluate the usefulness of LESS for end users, I conducted a survey on service creation by end users. The survey shows that relatively inexperienced users are willing and able to create their desired services, and our LESS-based service creation tool, Columbia University Telecommunication service Editor (CUTE), fits their needs.

Beyond end system service creation research, in this dissertation I also investigated location-based services in Internet telephony, specifically on Internet telephony end systems. I first analyze different location-based services in Internet telephony. Following the analysis, I introduce the implementation of the location-based services in our lab environment, as well as the prototype implementation of emergency call handling in SIP-based Internet telephony systems. Moreover, I also discuss our SIP-based global-scale ubiquitous computing architecture. The architecture uses the Service Location Protocol (SLP) to find available resources based on location information, then employs the SIP third-party call control architecture (3PCC) to control the available resources.

To test the hypotheses of my research, I have implemented a SIP user agent–SIPC. SIPC contains basic SIP functions as well as an end system service execution environment. It also supports many other Internet functions, such as service discovery, event notification, networked appliance control, instant messaging, and multicast media streaming based on the Session Announcement Protocol (SAP). Multiple functions integrated in SIPC can interact with each other to provide new services. This dissertation discusses the new services introduced by multi-function integration and interaction in detail.

Caption **Contest**

In what is sure to become an instant classic, the CUCS newsletter is delighted to kick off the very first installment of our Caption Contest.

The reader who provides the wittiest accompanying caption for this photograph of Professors Steve Bellovin and Angelos Keromytis (both experts in computer security) will win–yes, it's true–a year's free subscription to the CUCS newsletter.

Send your entry to: ***newslett@cs.columbia.edu***. We look forward to reading all responses and publishing the funniest ones!