CS@CU

NEWSLETTER OF THE DEPARTMENT OF COMPUTER SCIENCE AT COLUMBIA UNIVERSITY VOL.5 NO.1 WINTER 2008

The A-Z of Programming Languages: AWK

Professor Alfred V. Aho talks about the history and continuing popularity of his pattern matching language AWK.

Lawrence Gussman Professor of Computer Science Alfred V. Aho

The following article is reprinted with the permission of ***Computerworld Australia*** *(www.computerworld.com.au). The interview was conducted by Naomi Hamilton.*

Lawrence Gussman Professor of Computer Science **Alfred V. Aho** is a man at the forefront of computer science research. Formerly the Vice President of the Computing Sciences Research Center at Bell Labs, Professor Aho is well known for co-authoring the 'Dragon' book series and for being one of the three developers of the AWK pattern matching language in the mid-1970's, along with Brian Kernighan and Peter Weinberger. In the following interview, Professor Aho explains more about the development of AWK.

How did the idea/concept of the AWK language develop and come into practice?

As with a number of languages, it was born from the necessity to meet a need. As a researcher at Bell Labs in the early 1970s, I found myself keeping track of budgets, and keeping track of editorial correspondence. I was also teaching at a nearby university at the time, so I had to keep track of student grades as well.

I wanted to have a simple little language in which I could write one- or two-line programs to do these tasks. Brian Kernighan, a researcher next door to me at the Labs, also wanted to create a similar language. We had daily conversations which culminated in a desire to create a pattern-matching language suitable for simple data-processing tasks.

We were heavily influenced by GREP, a popular string-matching utility on UNIX, which had been created in our research center. GREP would search a file of text looking for lines matching a pattern consisting of a limited form of regular expressions, and then print all lines in the file that matched that regular expression.

We thought that we'd like to generalize the class of patterns to deal with numbers as well as strings. We also thought that we'd like to have more computational capability than just printing the line that matched the pattern.

So out of this grew AWK, a language based on the principle of pattern-action processing. It was built to do simple data processing: the ordinary data processing that we routinely did on a day-to-day basis. We just wanted to have a very simple scripting language that would allow us, and people who weren't very computer savvy, to be able to write throw-away programs for routine data processing.

Were there any programs or languages that already had these functions at the time you developed AWK?

Our original model was GREP. But GREP had a very limited form of pattern action processing, so we generalized the capabilities of GREP considerably. I was also interested at that time in string pattern matching

(continued on next page)

algorithms and context-free grammar parsing algorithms for compiler applications. This means that you can see a certain similarity between what AWK does and what the compiler construction tools LEX and YACC do.

LEX and YACC were tools that were built around string pattern matching algorithms that I was working on: LEX was designed to do lexical analysis and YACC syntax analysis. These tools were compiler construction utilities which were widely used in Bell labs, and later elsewhere, to create all sorts of little languages. Brian Kernighan was using them to make languages for typesetting mathematics and picture processing.

LEX is a tool that looks for lexemes in input text. Lexemes are sequences of characters that make up logical units. For example, a keyword like 'then' in a programming language is a lexeme. The character 't' by itself isn't interesting, 'h' by itself isn't interesting, but the combination 'then' is interesting. One of the first tasks a compiler has to do is read the source program and group its characters into lexemes.

AWK was influenced by this kind of textual processing, but AWK was aimed at data-processing tasks and it assumed very little background on the part of the user in terms of programming sophistication.

Can you provide our readers with a brief summary in your own words of AWK as a language?

AWK is a language for processing files of text. A file is treated as a sequence of records, and by default each line is a record. Each line is broken up into a sequence of fields, so we can think of the first word in a line as the first field, the second word as the second field, and so on. An AWK program is of a sequence of pattern-action statements. AWK reads the input a line at a time. A line is scanned for each pattern in the program, and for each pattern that matches, the associated action is executed.

A simple example should make this clear. Suppose we have a file in which each line is a name followed by a phone number. Let's say the file contains the line 'Naomi 1234'. In the AWK program the first field is referred to as $1, the second field as $2, and so on. Thus, we can create an AWK program to retrieve Naomi's phone number by simply writing $1=="Naomi" {print $2} which means if the first field matches Naomi, then print the second field. Now you're an AWK programmer! If you typed that program into AWK and presented it with a file that had names and phone numbers, then it would print 1234 as Naomi's phone number.

A typical AWK program would have several pattern-action statements. The patterns can be Boolean combinations of strings and numbers; the actions can be statements in a C-like programming language.

AWK became popular since it was one of the standard programs that came with every UNIX system.

What are you most proud of in the development of AWK?

AWK was developed by three people: me, Brian Kernighan and Peter Weinberger. Peter Weinberger was interested in what Brian and I were doing right from the start. We had created a grammatical specification for AWK but hadn't yet created the full run-time environment. Weinberger came along and said 'hey, this looks like a language I could use myself', and within a week he created a working run time for AWK. This initial form of AWK was very useful for writing the data processing routines that we were all interested in but more importantly it provided an evolvable platform for the language.

One of the most interesting parts of this project for me was that I got to know how Kernighan and Weinberger thought about language design: it was a really enlightening process! With the flexible compiler construction tools we had at our disposal, we very quickly evolved the language to adopt new useful syntactic and semantic constructs. We spent a whole year intensely debating what constructs should and shouldn't be in the language.

Language design is a very personal activity and each person brings to a language the classes of problems that they'd like to solve, and the manner in which they'd like them to be solved. I had a lot of fun creating AWK, and working with Kernighan and Weinberger was one of the most stimulating experiences of my career. I also learned I would not want to get into a programming contest with either of them however! Their programming abilities are formidable.

Interestingly, we did not intend the language to be used except by the three of us. But very quickly we discovered lots of other people had the need for the routine kind of data processing that AWK was good for. People didn't want to write hundred-line C programs to do data processing that could be done with a few lines of AWK, so lots of people started using AWK.

For many years AWK was one of the most popular commands on UNIX, and today, even though a number of other similar languages have come on the scene, AWK still ranks among the top 25 or 30 most popular programming languages in the world. And it all began as a little exercise to create a utility that the three of us would find useful for our own use.

How do you feel about AWK being so popular?

I am very happy that other people have found AWK useful. And not only did AWK attract a lot of users, other language designers later used it as a model for developing more powerful languages.

About 10 years after AWK was created, Larry Wall created a language called PERL, which was patterned after AWK and some other UNIX commands. PERL is now one of the most popular programming languages in the world. So not only was AWK popular when it was introduced but it also stimulated the creation of other popular languages.

AWK has inspired many other languages as you've already mentioned: why do you think this is?

What made AWK popular initially was its simplicity and the kinds of tasks it was built to do. It has a very simple programming model. The idea of pattern-action programming is very natural for people. We also made the language compatible with pipes in UNIX. The actions in AWK are really simple forms of C programs. You can write a simple action like {print $2} or you can write a much more complex C-like program as an action associated with a pattern. Some Wall Street financial houses used AWK when it first came out to balance their books because it was so easy to write data-processing programs in AWK.

AWK turned a number of people into programmers because the learning curve for the language was very shallow. Even today a large number of people continue to use AWK, saying languages such as PERL have become too complicated. Some say PERL has become such a complex language that it's become almost impossible to understand the programs once they've been written.

Another advantage of AWK is that the language is stable. We haven't changed it since the mid 1980's. And there are also lots of other people who've implemented versions of AWK on different platforms such as Windows.

How did you determine the order of initials in AWK?

This was not our choice. When our research colleagues saw the three of us in one or another's office, they'd walk by the open door and say 'AWK! AWK!'. So, we called the language AWK because of the good natured ribbing we received from our colleagues. We also thought it was a great name, and we put the AUK bird picture on the AWK book when we published it.

What did you learn from developing AWK that you still apply in your work today?

My research specialties include algorithms and programming languages. Many more people know me for AWK as they've used it personally. Fewer people know me for my theoretical papers even though they may be using the algorithms in them that have been implemented in various tools. One of the nice things about AWK is that it incorporates efficient string pattern matching algorithms that I was working on at the time we developed AWK. These pattern matching algorithms are also found in other UNIX utilities such as EGREP and FGREP, two string-matching tools I had written when I was experimenting with string pattern matching algorithms.

What AWK represents is a beautiful marriage of theory and practice. The best engineering is often built on top of a sound scientific foundation. In AWK we have taken expressive notations and efficient algorithms founded in computer science and engineered them to run well in practice.

I feel you gain wisdom by working with great people. Brian Kernighan is a master of useful programming language design. His basic precept of language design is to keep a language simple, so that a language is easy to understand and easy to use. I think this is great advice for any language designer.

Have you had any surprises in the way that AWK has developed over the years?

One Monday morning I walked into my office to find a person from the Bell Labs micro-electronics product division who had used AWK to create a multi-thousand-line computer-aided design system. I was just stunned. I thought that no one would ever write an AWK program with more than handful of statements. But he had written a powerful CAD development system in AWK because he could do it so quickly and with such facility. My biggest surprise is that AWK has been used in many different applications that none of us had initially envisaged. But perhaps that's the sign of a good tool, as you use a screwdriver for many more things than turning screws.

Do you still work with AWK today?

Since it's so useful for routine data processing I use it daily. For example, I use it whenever I'm writing papers and books. Because it has associative arrays, I have a simple two-line AWK program that translates symbolically named figures and examples into numerically encoded figures and examples; for instance, it translates 'Figure AWK-program' into 'Figure 1.1'. This AWK program allows me to rearrange and renumber figures and examples at will in my papers and books. I once saw a paper that had a 1000-line C program that had less functionality than these two lines of AWK. The economy of expression you can get from AWK can be very impressive.

How has being one of the three creators of AWK impacted your career?

As I said, many programmers know me for AWK, but the computer science research community is much more familiar with my theoretical work. So I initially viewed the creation of AWK as a learning experience and a diversion rather than part of my regular research activities. However, the experience of implementing AWK has greatly influenced how I now teach programming languages and compilers, and software engineering.

What I've noticed is that some scientists aren't as well known for their primary field of research by the world at large as they are for their useful tools. Don Knuth, for example, is one of the world's foremost computer scientists, a founder of the field of computer algorithms. However, he developed a language for typesetting technical papers, called TEX. This wasn't his main avenue of research but TEX became very widely used throughout the world by many scientists outside of computer science. Knuth was passionate about having a mathematical typesetting system that could be used to produce beautiful looking papers and books.

Many other computer science researchers have developed useful programming languages as a by-product of their main line of research as well. As another example, Bjarne Stroustrup developed the widely used C++ programming language because he wanted to write network simulators.

Would you do anything differently in the development of AWK looking back?

One of the things that I would have done differently is instituting rigorous testing as we started to develop the language. We initially created AWK as a 'throw-away' language, so we didn't do rigorous quality control as part of our initial implementation.

I mentioned to you earlier that there was a person who wrote a CAD system in AWK. The reason he initially came to see me was to report a bug in the AWK complier. He was very testy with me saying I had wasted three weeks of his life, as he had been looking for a bug in his own code only to discover that it was a bug in the AWK compiler! I huddled with Brian Kernighan after this, and we agreed we really need to do something differently in terms of quality control. So we instituted a rigorous regression test for all of the features of AWK. Any of the three of us who put a new feature into the language from then on, first had to write a test for the new feature.

I have been teaching the programming languages and compilers course at Columbia University for several years. The course has a semester long project in which students work in teams of four or five to design their own innovative little language and to make a compiler for it.

Students coming into the course have never looked inside a compiler before, but in all the years I've been teaching this course, never has a team failed to deliver a working compiler at the end of the course. All of this is due to the experience I had in developing AWK with Kernighan and Weinberger. In addition to learning the principles of language and compiler design, the students learn good software engineering practices. Rigorous testing is something students do from the start. The students also learn the elements of project management, teamwork, and communication skills, both oral and written. So from that perspective AWK has significantly influenced how I teach programming languages and compilers and software development.

Message from the Chair

Henning Schulzrinne, Professor & Chair

For several years now, low enrollments in computer science have provided plenty of fodder for panels, editorials and faculty lunches. Right now there are some mildly encouraging signs; at Columbia University, "competing" majors such as financial engineering may be drawing fewer students, and there seems to be some progress in convincing the public that most CS-related jobs have not shipped to Bangalore. However, it seems unlikely that we will be returning to the heady enrollment numbers of a decade ago any time soon. I also believe that simply relying on better public relations or the decreasing attractiveness of Wall Street jobs alone is not quite enough.

Among many other aspects, such as a more relevant curriculum, I believe that professionalizing computer science may contribute to a better reputation of the field. Right now, the public understandably considers vastly different computing-related fields and occupations as "working with computers", ranging from first-line technical support reading from canned scripts to software engineers designing new algorithms for computational biology. Even the various classifications of the Bureau of Labor Statistics (BLS) conflate very different backgrounds. As one example, the BLS notes that 20% of computer programmers have no degree, roughly half have a bachelor's degree and 20% have completed a graduate degree, although it is not clear how many of these are indeed in computer science or computer engineering. Among the five occupations most closely related to computer science, all but one (computer software engineer) list associate to graduate degree as prerequisites, a far wider spectrum than one is likely to find for other engineering professions, let alone the classical professions such as lawyers or architects.

Unfortunately, it is not clear how the muddled picture of the computer profession can be clarified, given the lack of professional certification equivalent to, say, the CPA or bar exam. However, I believe that we can do more to install a professional ethic in our students. Deborah Johnson describes, in the October CACM, the fiduciary model as a promising model to highlight the difference between guns-for-hire and professionals. Making students think of themselves as professionals-in-training may also help. Below I highlight three aspects of such an attitude: emphasis on quality, public service, and professional engagement.

In far too many cases, student projects seem to be motivated by the "can I get it by the TA" and "will it survive a demo" tests, rather than pride in professional-quality work. There seems to often be little stomach to take off points for sloppy code or lackluster documentation, as long as the results are correct. Judging from some of the projects produced by our MS students (most of whom come from other institutions), this does not seem to be a Columbia-specific issue. In one telling remark, a student of mine who worked on the FAA project described in this issue noted how different it was to program for class and semester projects compared to a project used 12 hours a day by non-computer scientists. In contrast to other professions, we seem to tolerate lower quality for our trainees than what's expected in the field. Emphasis, from the very beginning, on verification, testing and sound engineering principles may help to set apart the professionals we hope to train from the amateurs who picked up a bit of programming from "Java for Dummies". Just knowing O() notation is not sufficient.

Most professions have a tradition of public service, with engineering and computer science probably weaker than most. Just as law students run clinics, we may consider ways to increase the participation of CS students (and faculty...) in projects that benefit the public, not just through research. At all levels, from local government to the Red Cross, contributions of trained or in-training computer scientists could dramatically contribute to the public welfare. I suspect it would be easy to find interested students and faculty, given the real-world impact such projects can have, but matching up governments, not-for-profits and universities seems challenging and time-consuming. At Columbia University, our Design Fundamentals course required for all freshmen includes community projects, but they are limited in scope by the relative inexperience of the students. Professional societies such as ACM and CRA might be able to increase awareness on both sides. Despite the difficulties, I am convinced that showing students, through their own experience, that computer science can make a real difference in the world at large can only help to make the field more attractive. I look forward to hearing your perspectives. As always, I look forward to hearing from you as our readers.

New **Faculty**

Computer Science Department Welcomes **Professor Martha Mercaldi Kim**

Professor **Martha Mercaldi Kim**

The Computer Science department is delighted to welcome Martha Mercaldi Kim as an Assistant Professor of Computer Science.

Martha Kim's research focus is computer architecture. Her particular interest is in problems that lie at the boundaries of architecture. The rapid scaling of modern silicon processes has stretched the traditional circuit-architecture and architecture-software abstractions to their breaking points. Architectures cannot function while ignoring circuit-level effects. Nor can architectures any longer single-handedly harness growing computational resources and leave appearances to software unchanged.

Martha's research has explored problems such as these. She has investigated compilation algorithms for mapping application software to modern chips that contain many parallel computational cores. In the purely hardware domain, she has designed a low-cost style of chip, called brick and mortar, with the goal of reaping the benefits of modern circuit integration without the financial side effects. Brick and mortar chips are assembled from multiple small, pre-fabricated hardware components (the bricks) that are bonded in a designer-specified arrangement to a communication backbone chip (the mortar). Stemming from this research she has identified the need for configurable on-chip communication, not only in brick and mortar chips but in traditional chips as well. Recently she has proposed and evaluated a design of such a polymorphic on-chip network.

Martha received her Ph.D. in Computer Science and Engineering from the University of Washington in December 2008. Prior to that she earned her bachelors in Computer Science from Harvard University in 2002 and a masters in embedded systems design from the University of Lugano in Switzerland in 2003.

Faculty Award

Ken Ross Receives **Distinguished Faculty Teaching Award**

Professor **Ken Ross**

Computer Science Professor **Ken Ross** was selected as one of the two recipients of this year's Columbia Engineering School Alumni Association (CESAA) Distinguished Faculty Teaching Awards.

The other recipient was Professor David Keyes of the Department of Applied Physics and Applied Mathematics.

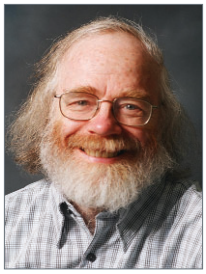
Professor Ross was recognized for his courses in databases and problem solving. Mr. Lee presented the award to Professor Ross and Professor Keyes at Class Day ceremonies on Monday, May 19. "The Columbia Engineering School Alumni Association created this award more than a decade ago to recognize the exceptional commitment of members of the SEAS faculty to undergraduate education," said Mr. Lee. "This year, I am pleased to present these awards to two senior faculty members, a testament to their continuing faithfulness to the central mission of teaching undergraduates."

The awardees were selected by a Committee of the Alumni Association chaired by Eric Schon '68, with representation from the student body, and based on nominations from the students themselves. The Board of Managers of the Columbia Engineering School Alumni Association voted unanimously to approve the selection.

Students enthusiastically wrote that courses taught by these professors were the best they have taken at Columbia. The qualities that both professors share and the ones most frequently mentioned by students are their enthusiasm for the subject matter, caring attitude, approachability, responsiveness to student concerns, and the ability to make complex subject matter understandable.

CUCS Faculty Receive **Google Research Awards**

Professor
Peter Allen



Professor
Steven Bellovin

Professor
Angelos Keromytis

Professor
Rocco Servedio

Professor
Sal Stolfo

Professors **Peter Allen**, **Steven Bellovin**, **Angelos Keromytis**, **Rocco Servedio** and **Sal Stolfo** have all recently received Google research awards for projects on 3D databases, safe browsing, anonymous networking, privacy-preserving web searching and martingale ranking algorithms.

Professor **Peter Allen** will be investigating semantically searchable dynamic 3D databases, developing new methods to take an unstructured set of 3D models and organize them into a database that can be intelligently and efficiently queried. The database will be searchable, tagged and dynamic, and will be able to support queries based on whole object and partial object geometries.

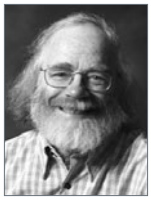
In the project titled "Safe Browsing Through Web-based Application Communities", Professors **Angelos Keromytis** and **Sal Stolfo** will investigate the use of collaborative software monitoring, anomaly detection, and software self-healing to enable groups of users to browse safely. The project seeks to counter the increasingly virulent class of web-bourne malware by exchanging information among users about detected attacks and countermeasures when browsing unknown websites or even specific pages.

In the project "Privacy and Search: Having it Both Ways in Web Services", Professor **Angelos Keromytis** will investigate techniques for addressing the privacy and confidentiality concerns of businesses and individuals while allowing for the use of hosted, web-based applications such as Google Docs and Gmail. Specifically, the project will combine data confidentiality mechanisms with Private Information Matching and Retrieval protocols, to develop schemes that offer different tradeoffs between stored-data confidentiality/privacy and legitimate business and user needs.

Professor **Rocco Servedio** was awarded a Google Research Award to develop improved martingale ranking algorithms. Martingale ranking is an extension of martingale boosting, a provably noise-tolerant boosting algorithm from learning theory which was jointly developed by Prof. Servedio and Phil Long, a researcher at Google. A goal of the project is to design adaptive and noise-tolerant martingale rankers that perform well 'at the top of the list' of items being ranked, which is where accurate rankings are most important.

The Physical World **and the Real World**

The following piece by Professor Steven Bellovin appeared as an "Inside Risks" article in the May 2008 issue of ***Communications of the ACM****.*



Professor
Steven Bellovin

Most of us rely on the Internet, for news, entertainment, research, communication with our families, friends, and colleagues, and more. What if it went away?

Precisely that happened to many people in early February, in the wake of the failure of several undersea cables. According to some reports, more than 80 million users were affected by the outages, Both the failure and the recovery have lessons to teach us.

The first lesson, of course, is that failures happen. In fact, multiple failures can happen. Simply having some redundancy may not be sufficient; one needs to have enough redundancy, of the right types. In this case, geography and politics made life tougher.

The geographical issue is clear from looking at a map: there aren't many good choices for an all-water route between Europe and the Persian Gulf or India. And despite this series of events, cables are generally thought to be safer on the seabed than on land. (A standing joke in the network operator community holds that you should bring a length of fiber optic cable with you when going hiking in the wilderness. If you get lost, throw it on the ground. A backhoe will soon show up to sever it; ask the driver how to get home.)

The obvious answer is to run some backup cables on land, bypassing the chokepoint of the Red Sea. Again, a glance at the map shows how few choices there are. Bypassing the Red Sea on the west would require routing through very unstable countries. An eastern bypass would require cooperation from mutually hostile countries. Neither choice is attractive.

From this perspective, it doesn't matter much just why the cables failed. Cables can be cut by ship anchors, fishing trawlers, earthquakes, hostile action, even shark bites. Regardless of the cause, when so many cables are in such a small area, the failure modes are no longer independent.

For this problem, there are no good solutions. Anyone whose business depends on connectivity through this region must take this into account.

The dangers aren't only physical. The last few months have also shown that a 1999 National Research Council report was quite correct when it warned of the fragility of the routing system and the domain name system.

In one highly-publicized incident, a routing mistake by a Pakistani ISP knocked Youtube off the air. There was a lot of speculation that this was deliberate—the government of Pakistan had ordered Youtube banned within the country; might someone have tried to "ban" it globally?—though later analysis strongly suggests that it was an innocent mistake. An outage affecting such a popular site is very noticeable; there was a great deal of press coverage. By contrast, when a Kenyan network was inadvertently hijacked by an American ISP, there was virtually no notice. Quieter, deliberate misrouting—say, to eavesdrop on traffic to or from a small site—might go completely unnoticed.

The DNS-related incidents are scarier because they do reflect deliberate actions, with the force of the U.S. legal system behind them. In one case, the Wikileaks.org web site was briefly deleted from the DNS by court order, because a bank claimed the site contained stolen documents. (The site owners had apparently foreseen something like that, and had registered other names for the site in other countries: the .org registry is located in the U.S.) In a second incident, a U.S. government agency ordered the names of some non-U.S. sites removed from .com (again, located in the U.S.) because they violated the embargo against Cuba.

What can we learn from these incidents? The moral is simple: the Internet is a lot more fragile than it appears. Most of the time, it works and works very well, without government interference, routing mistakes, or outages due to occasional fiber cuts. Sometimes, though, things go badly wrong. Prudence dictates that we plan for such instances.

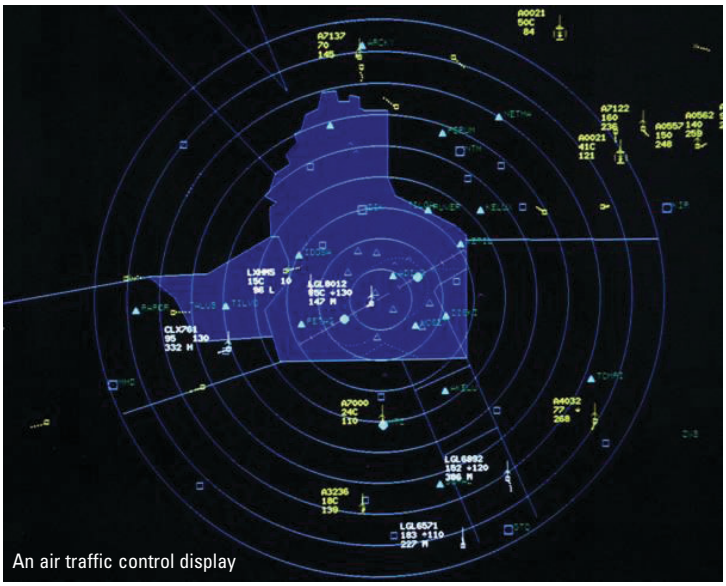
Training air traffic controllers **using VoIP**

Professor **Henning Schulzrinne**

Voice-over-IP has become a well-established technology and is well on its way to replace the traditional circuit-switched phone services, both within the enterprise and for the public phone system.

As part of a research project with the Federal Aviation Administration (FAA), the Internet Real-Time (IRT) Laboratory explored how to use these technologies for a somewhat unusual purpose: training future air traffic controllers in radio communications skills. In scripted training exercises, students learn how to communicate with (actor) pilots and other air traffic controllers (ATCs), interacting with user interfaces that approximate the technology they will find in a control tower. Depending on the scenario, small groups of air traffic controllers and pilots need to be linked together. Unlike most research prototypes, the software must survive 12 hours a day of classroom exercises, with changing sets of instructors. Thousands of ATCs need to be trained in the next few years, as the generation of controllers hired after the 1981 strike retires.

The project has its origin in 2004, when one of the training managers of the FAA Academy in Oklahoma City, Susan Buriak, felt that they could no longer continue using the hard-wired analog communication system which had been in use for many years in their training facility. Its hardware was becoming obsolete and any custom manufacturing would have been prohibitively expensive, so the FAA needed an alternative. The FAA manager had read about VoIP and contacted Prof. Henning Schulzrinne to see if technology developed at Columbia could be repurposed to build a modern training facility.



An air traffic control display

The training rooms at the FAA Academy had three parallel networks: a data network, a voice network and a video network. VoIP could unify all communications on a single Ethernet and allow to re-configure the classroom setting through a control panel. Radio links were modeled using IP multicast, while tower-to-tower communications used point-to-point VoIP sessions. The instructor has to be able to listen in on on-going exercises; recordings are used to evaluate the performance of ATC trainees. The system is build on most of the protocols developed in the IRT lab, including SIP, RTSP and RTP. Several exercises are typically run in parallel and must not interfere with each other.

To increase verisimilitude with the systems found in a control tower, the simulator includes many of the same user interface components, such as a push-to-talk microphone, a foot pedal and touch screens that can be reconfigured to approximate various console configurations.

Designing a system in an unknown domain, aviation training, mandated the use of alternate software development methodologies—rapid prototyping and iterative development—and as the prototypes went back and forth between the FAA and Columbia, the IRT researchers built up significant domain expertise. A uniform wrapper-based interface efficiently handles the large number of I/O devices and eases the integration of newer I/O devices. Similar to industrial control systems, self-correcting mechanisms were added that employed heart-beat checks to restart any non-responding components and restore its status prior to the fault. The project initially started with one facility, but the system has since been deployed and operational in five FAA Academy training rooms since late 2006, with two more expected by the end of 2008. Except for supporting software components such as operating systems, web servers and SQL database, all core components are based on software developed by IRT students and staff, often extending or modifying work done earlier for other projects.

In the course of the project, students have made five on-site visits to the FAA facility in Oklahoma City and have taught FAA technical staff in two training sessions at Columbia University.

Designing such a system within the resource constraints of a university lab posed special challenges, especially over the last two years, after the FAA went ahead with the production system deployment. The system being deployed in an environment where none of the users are CS majors, has meant that even trivial problems need administrator intervention—with a few of those being escalated all the way to the Washington DC headquarters! With remote debugging proving to be daunting and stoppage of ATC training having legal implications, the IRT team had to resort to on-site visits, even for seemingly trivial issues. Problems with the associated hardware (e.g., firmware issues in a push-to-talk headset) and occasional bugs in the VoIP system, did not prove helpful either. Needless to say, the project team has learned how important extensive logging and fail-safe mechanisms are, as well as having well-trained local system administrators.

Encouraged by the potential of the initial VoIP system, the FAA funded a subsequent effort to design an advanced VoIP system to be used in the high-fidelity lab environments. While the system is for training only, the on-going FAA Next Generation Air Transportation System (NextGen) is also likely to use VoIP, albeit developed by much larger teams and probably not in a university.

Next time you listen in on air traffic radio on channel 9 on United Airlines, you may well be listening to a controller trained on CUCS software...

The **Emerging Scholar's Program**

Ph.D. Student
Chris Murphy

Ph.D. Student
Kristen Parton

Professor
Adam Cannon

ESP peer leader and junior CS major Sahar Hasan helps students complete a decision tree for the 9 coins problem

Did you know that from 1984 to 2007 the percentage of CS degrees awarded to women nationwide has actually decreased, from 37% to 19%[1]?

In the Columbia Computer Science Department last year, 18% of undergraduates were women, as compared with 37% of all SEAS undergraduates. To try to counter this trend at Columbia, Ph.D. students Chris Murphy and Kristen Parton and Computer Science Professor Adam Cannon started the Emerging Scholars Program (ESP), a peer-led workshop designed to encourage talented female students to stay in the CS major after introductory CS classes.

The goal of ESP is to show students that CS is necessarily a collaborative activity and that it involves much more than just programming. It is targeted towards female students enrolled in Introduction to Computer Science classes who have not yet declared a major, but are contemplating CS. Each week, small groups meet with a peer leader who presents one or more problems from a variety of CS fields, from robotics to natural language processing. Students work as a group to analyze the problems and brainstorm solutions. Since there is no programming and no homework, students can focus on algorithmic thinking and analyzing problems to see that there is more to CS than just programming.

Professor Kathleen McKeown has noticed the scarcity of undergraduate women in the courses she teaches, such as Artificial Intelligence. "Discussing how to approach and solve these problems without programming will let these women see early on why Computer Science is fun. I think ESP can give them an edge and sense of confidence to combat the feeling that the men know more." Another benefit of ESP is improved networking opportunities for beginning CS students. Since many later CS courses involve group projects, it is important to get to know peers in the major early on. As one ESP participant said, "Women make up a small part of the CS major so it's nice to meet other girls who might share the interest."

Last semester, ESP started as a pilot program, supported by Women in Computer Science (WICS), with six students from the Introduction to Computer Science course. CS major Kim Manis led weekly hour-long workshops, with help from workshop assistant and CS major Sahar Hasan. After a successful pilot program, ESP was awarded a $15,000 two-year grant from the National Center for Women in Information Technology and Microsoft Research. With this grant, as well as strong support from the Columbia CS department, ESP has expanded to two sections this Fall, with a total of 15 enrolled students.

ESP follows the Peer-Led Team Learning teaching paradigm, which has been successfully used in Computer Science departments at Duke University, Georgia Tech, Rutgers University and others. Since the workshops are led by another student rather than a professor or TA, the sessions can be more interactive. Participant Jana Johnson said, "I loved being in a small group because it forced us all to participate and I also loved having a workshop leader that was close in age and encouraged creative thinking." Johnson has continued with ESP this semester as a workshop assistant, along with sophomore Elba Garza, who also participated last semester. Hasan, who started in the spring as a workshop assistant, is continuing this fall as a peer leader, along with junior CS major Shylah Weber.

Although the workshops are led by an undergraduate peer leader, the classroom materials are prepared by Ph.D. student coordinators in conjunction with Professor Cannon. One of the favorite sessions last semester was a social networking workshop, which used friendship networks from *Facebook.com* to explore graph algorithms, such as finding friends of friends and detecting cliques. ESP coordinator Murphy said, "It's challenging to come up with interesting and fun material, especially so that the workshops are more interactive and don't turn into lectures. The feedback we get from the students has really helped us understand what sorts of topics they're interested in."

The program received very positive feedback at the end of the semester, with all students saying they would strongly recommend it to another student. ESP student Diana Cimino explained: "Through the varied workshops, I was exposed to interesting people and ideas, realizing the breadth of an entirely fascinating subject in which I had no previous experience."

For more information about ESP, see http://www.cs.columbia.edu/esp/

[1] National Center for Women & Information Technology Fact Sheet. *http://www.ncwit.org/about.factsheet.html*

Apple2fpga: Reconstructing an Apple II+ on an FPGA

Professor **Stephen Edwards**

As a Christmas present to myself in 2007, I implemented an 1980s-era Apple II+ in VHDL to run on an Altera DE2 FPGA board.

The point, aside from entertainment, was to illustrate the power (or rather, low power) of modern FPGAs. Put another way, what made Steve Jobs his first million can now be a class project (well, almost) for my 4840 embedded systems class.

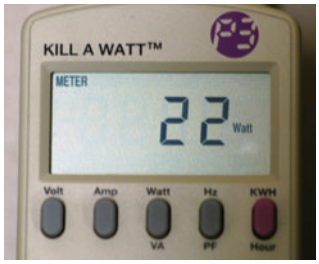
What is an Apple II?

The Apple II was one of the first really successful personal computers. Designed by Steve Wozniak ("Woz") and first introduced in 1977, it really took off in 1978 when the 140K Disk II 5.25-inch floppy drive was introduced, followed by VisiCalc, the first spreadsheet program.

Fairly simple even by the standards of the day, the Apple II was built around the inexpensive 8-bit 6502 processor from MOS Technology (it sold for $25 when an Intel 8080 sold for $179). The 6502 had an eight-bit data bus and could access 64K of memory. In the Apple II, it runs at slightly above 1 MHz. Aside from the ROMs and DRAMs, the rest of the circuitry consisted of discrete LS TTL chips.

While the first Apple IIs shipped with 4K of DRAM, this quickly grew to a standard of 48K and later to 64K through the help of bank switching and a memory expansion card. DRAMs, at this time, were cutting-edge technology. While they were difficult to use, requiring +5, +12, and -5V power supplies and needing to be periodically refreshed, their roughly six-times improvement in density over more standard static memory chips made it worthwhile. Breathless copy in an early ad for the Apple I explicitly touted the advantages of DRAM chips.

While the Apple II came with an integrated keyboard, a rudimentary (one-bit) sound port, and a game port typically connected to a two-axis analog joystick, its main feature was its integrated video display. It generated composite (baseband) NTSC video that was usually sent through an RF modulator to appear on e.g., a TV's channel 3.



The original system: Apple II+; c.1982; Synertek 6502; LS TTL, 16K DRAMs, 1 MHz NMOS CPU; CPU: 4K transistors?; 22 Watts, 31 when disk spinning

The Apple II had three video modes: a 40×24 uppercase-only black-and-white text display, a 40×48 16-color low-resolution display, and a 140×280 6-color high-resolution display.

The Apple II can almost be thought of as a video display device that happens to have a microprocessor connected to it. Woz started with a 14.31818 MHz master pixel clock—exactly four times the 3.579545 MHz colorburst frequency used in NTSC video. Because of the way color was added to the black-and-white NTSC standard, seemingly black-and-white patterns sent at exactly this frequency are interpreted as color.

Woz derived the CPU clock from the 14.31818 MHz clock by dividing by roughly fourteen. I say roughly because in fact every sixty-fifth CPU cycle (once per horizontal scan line) is stretched by two 14 MHz clock periods to preserve the phase of the 3.58 MHz colorburst frequency. Thus, there are 65*14+2=912 pixel periods per line, or exactly 228 cycles of the 3.58 MHz colorburst per line.

Both the CPU and video access a byte of memory at 1 MHz. Their accesses are interleaved, so the DRAM effectively operates at 2 MHz. Another Woz trick: the video addresses are such that refreshing the video also suffices to refresh the DRAMs, so no additional refresh cycles are needed.

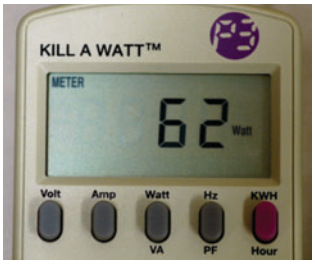
In my reconstruction, I tried to reproduce the behavior of the timing circuitry (including the stretched cycle) as closely as possible. However, rather than actually generate lower-frequency clocks in this way (generally a bad idea in modern designs, where the use of on-chip PLLs with skew control is the preferred technique), I ran most of the core at the 14 MHz clock and distributed latch enable signals to anything that was originally clocked at one of the derived frequencies.

My Reconstruction

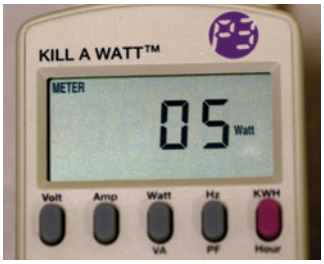
My Apple II core consists of a timing generator, a video generator, the 6502 processor core, which I took from Peter Wendrich's Commodore 64 emulator, the ROMS, and some random logic for address decoding and other onboard I/O. Broadly, the core expects a 14.31818 MHz clock signal, input from the keyboard, access to a 64K RAM, and access to the peripheral bus (currently, just a disk drive emulator) and generates a one-bit video stream along with one bit of audio data for the speaker.

To make this core usable, I attached to it a PS/2 keyboard controller (from Alex Freed), a disk emulator, and a VGA line-doubler that converts the 15 kHz horizontal refresh rate of the Apple's one-bit video output to a color VGA output with 30 kHz horizontal refresh rate.

The line doubler contains memory for two lines of video from the Apple; I display one twice while the other is being read from the Apple's video out. In addition to interpreting and generating the horizontal and vertical synchronization signals, it converts the one-bit video stream to color using an overly-simple algorithm: every four 14 MHz pixels are interpreted as a block and displayed as a



The system emulated in software: Dell Optiplex Gxa; c.1998; Intel Pentium II; 233 MHz, 250 nm CMOS CPU; CPU: 7.5M transistors; 62 Watts



The system emulated on an FPGA: Altera/Terasic DE2; c.2006; Altera EP2C35F672C6 Cyclone II; 90 nm CMOS; 33K LEs, 150M transistors?; 5 Watts

single color. This is reasonably effective, and produces some of the odd color fringing effects of the original Apple II, but is not quite right since a true television set actually performs more gentle low-pass filtering on the luminance and chrominance signals. A better algorithm would consider bits adjacent to each group of four. Nevertheless, the display is certainly usable.

My emulation of the Disk II is primitive (read-only) but functional. I store a "nibblized" image of a 5.25-inch floppy on an external SD card and read it into on-chip memory a track at a time. I say "nibblized" because like most disk drives, the Disk II doesn't directly store binary data but instead encodes it to limit the maximum length with no magnetic polarity changes, which makes it possible to recover the bit clock. Since the Apple II decodes this bitstream in software, my disk controller delivers the raw (encoded) bitstream. Although this wastes flash storage space, an encoded 140K disk image only takes 228K. With 2 Gb SD cards selling for less than $10 as of late 2008, it hardly matters.

I coded an SPI interface (one of the three spoken by typical SD cards) in VHDL that is able to wake up the card and download random 256-byte blocks. Controlling this is a simple emulator for the Disk II controller, which provides a low-level interface to the drive. For example, the read/write head is moved under software control by selectively pulsing four stepper motor phases. My hardware emulator watches these pulses and follows which track the head currently resides. Each time the track number changes, my controller reads from the SD card a new group of 6655 bytes (256*25) into track memory. Meanwhile, I emulate the spinning disk by periodically changing the address of the data read by the CPU when it accesses the disk I/O locations.

For debugging, I brought out the CPU's PC to four of the seven-segment LED displays on the board and the current track for the drive on another two. While the PC is usually changing so fast it becomes a blur, patterns do often emerge. For example, the PC remains in a small range when the computer is waiting at the prompt. Similarly, I have found a lot of software, including the operating system when it is moving the drive head, calls the monitor's "delay" routine to slow things down.

Comparing Implementations

A central motivation for this project was to illustrate how integrated modern FPGAs have become and how little power they can consume. So I compared the power consumed by an actual Apple II+, an Apple II+ emulated in software, and my FPGA reconstruction.

To measure power, I used the inexpensive and very easy to use "Kill A Watt" power meter. This claims 0.2% accuracy, which is plenty to get a rough idea of what (watt?) is going on.

In each case, I measured the power consumed by the system excluding the monitor. The power was more-or-less constant for the Dell (while the emulator was running) and FPGA board; the Apple II's power consumption varied considerably when the disk was active (spinning).

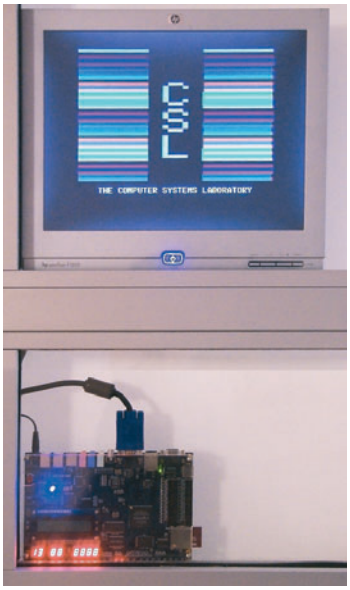
Only the number for the Apple II is really fair. The Dell is freakishly overpowered, had far more memory (192 MB) than necessary to run Linux and the Apple II emulator, and never used its floppy drive, CD ROM, etc. The FPGA board has similar problems: it also has (unused) Ethernet, USB, and NTSC video interfaces as well as SDRAM and Flash chips. While none were running, they still consumed some power.

Conclusions

My FPGA version of an Apple II+ succeeded in entertaining me and demonstrating something to my 4840 class, but has had additional impact. Alex Freed, whose previous Apple II-on-an-FPGA inspired my own, went back and further improved his design, and it got some play on the "reddit" website, in which people vote for interesting things they see on the Internet. It even prompted a job offer (relax; I'm still at Columbia). Finally, I'm using it to power the sign for our Computer Systems Laboratory, which we wrote about in an earlier newsletter.

My Apple II+ reconstruction is one in a growing collection of historical computer emulators available on the Internet. Motivated by nostalgia as much as historical presentation, software emulators for virtually every successful computer platform have already been written (e.g., see the MESS and SIMH projects) as well as for virtually every video arcade game (see the MAME project) and videogame console. Apple itself has used such emulation technology to allow, e.g., Mac OS 9 programs to run under Mac OS X.

There are fewer historical hardware emulation projects, owing largely to the need for FPGA hardware, but a variety of video arcade games as well as older home computers have also been emulated (such as the VIC-20; *see fpgaarcade.com and pacedev.net*). One of my hopes is that my project will encourage others to reproduce their favorite systems in a more future-friendly format.



The Apple II-on-an-FPGA controlling the sign for the Computer Systems Laboratory.

New **Faces**

Mohamed Altantawy graduated from The American University in Cairo with a B.S. in Computer Science in 2005. He worked at IBM Egypt from 2005-2006. Mohamed did an M.S. in computer science at The American University in Cairo from 2006-2008. He started as a Ph.D. student in fall 2008. His main research interests are Computational Linguistics and Social Networks. He is currently working with Dr. Owen Rambow in the Center for Computational Learning Systems.

Ang Cui graduated from Columbia University with a B.S in Computer Science in 2005. After a three year network administration adventure at D.E. Shaw & Co, he returns to Columbia as a Ph.D. student. Ang's research interests include network security and intrusion detection systems. He is currently working in Prof. Stolfo's IDS lab. In his free time, Ang plays the erhu and aspires to become a competetive cyclist.

Joshua B. Gordon graduated from Columbia with a B.S. in Computer Science in December 2006. He returned to pursue a Ph.D. in Fall 2008 after working at Lawrence Livermore National Laboratory. His research interests are in artificial intelligence and natural language. He is currently working with Dr. Passonneau at the Center for Computational Learning Systems.

Ragesh Jaiswal grew up in India, where he received his undergraduate degree in Computer Science and Engineering at the Indian Institute of Technology Kanpur in 2003. He moved to San Diego, CA to pursue a Ph.D. in Computer Science and Engineering at the University of California San Diego under the supervision of Prof. Russell Impagliazzo. He graduated from UC San Diego in 2008. His doctoral research work focuses on Direct Product Theorems and their applications in Cryptography, Derandomization, Computational Complexity, and Error correcting codes. He is now a postdoc at Columbia working with Prof. Rocco Servedio.

Kangkook Jee graduated from Korea University with a B.S. in Mathematics in 2001. Afterwards, he worked at IBM Korea for five years. Kangkook did an M.S in computer science at Columbia from 2006-2007 and started as a Ph.D. student in Spring 2008. His main research interests are operating systems and system security. He is currently working with Prof. Angelos Keromytis's NSL (Network Security Laboratory) group.

Vasileios Kemerlis graduated in 2006 from Athens University of Economics and Business with a B.Sc. in Computer Science. During his undergraduate studies he held appointments with the same department as a network administrator, lab assistant and, after his graduation, as a research fellow working in the areas of wireless networks and data integration in mobile environments. During 2007-2008 he was with Mysapient Ltd (UK) as a software engineer and at the same time served as a security consultant at the Hellenic National Defense General Staff, department of cyber defense. While an undergraduate student he was also a member of Microsoft's developers platform evangelists group. In 2007 he was awarded the Ericsson's award of Excellence in Telecommunications for his B.Sc. thesis. Vasileios joined the Ph.D. program at Columbia in Fall 2008 and currently he is working with Prof. Angelos Keromytis in network security.

Ahmed M. El Kholy graduated from The American University in Cairo (AUC) with a B.Sc. in Computer Science in June 2005. He worked at IBM Corporation from 2005-2006 and 2008, did a M.Sc. in computer science at the same University (AUC) from 2006-2008, and started as a Ph.D. student at Columbia University this Fall 2008. Ahmed's main research interests are Natural Language Processing, Social Networking and Machine Learning. He is currently working with Dr. Nizar Habash at The Center for Computational Learning Systems (CCLS).

Sungjun Kim graduated from Seoul National University with a B.S. in Electrical Engineering (Major) and in Business Administration (Double Major) in February 2002. He worked at BIOMEDLAB Corporation in 2002, HUNEED Corporation in 2003-4, and TOPFIELD Corporation in 2004-6, and started as a Ph.D. student in Spring 2008. Sungjun's main research interest is Embedded Systems. He is currently working with Prof. Stephen A. Edwards's group, and is involved in the Precision Timed (PRET) project.

Troy Lee graduated from Caltech with a B.S. in mathematics in 2000. He then did his Masters and Ph.D. in Computer Science at the University of Amsterdam, finishing in 2006. Before becoming a postdoc at Columbia, where he is working with Prof. Rocco Servedio, he was a postdoc at University Paris-Sud and Rutgers University. His main research interests are in computational complexity, in particular communication complexity and quantum computing.

Wei-Yun Ma graduated from Columbia University with a M.S. in Computer Science in 2008. Before coming to Columbia in 2006, he had worked at Institute of Information Science, Academia Sinica, Taiwan for five years. He started his Ph.D. studies at Columbia in Fall 2008. He is working with Prof. Kathy McKeown in the Natural Language Processing Lab. His research interests include Natural Language Processing, Information Retrieval and Machine Learning. He is currently working on improving machine translation for a multilingual QA system.

Snehit Prabhu hails from a small seaside town in Goa, India. He finished his undergrad education in Bangalore in 2003, where he majored in Computer Science. He subsequently spent a few years working at IBM Software Labs and Research, where he dabbled in Bioinformatics, Autonomic Computing and Web Services among other things. He joined the Ph.D. program at Columbia in Spring 2008. His

primary research is in Theoretical and Computational Biology: particularly, understanding the role and contribution of genetic variation in evolution. He works at the Computational Genetics Lab headed by Prof. Itsik Pe'er, and is also advised by Prof. Yechiam Yemini.

Baolin Shao graduated from School of Software Engineering, Huazhong University of Science and Technology with a B.E. in Software Engineering in June 2007. Before that, he held an internship at Infosys Technologies Ltd. from September 2006 to April 2007 in Bangalore, India. He started as a Ph.D. Student this fall. Baolin's main research interests are computer language and compilers, and formal verification. He is currently working with Prof. Stephen A. Edwards on the SHIM project.

Swapneel K. Sheth graduated from Sardar Patel College of Engineering, Mumbai, India with a B.E. in Computer Engineering in May 2006. He did an M.S. in Computer Science at Columbia University from 2006 to 2007 and started as a Ph.D. student in Spring 2008. Swapneel's main research interest is Software Systems. He is currently working with Prof. Gail Kaiser on using social networking metaphors for knowledge sharing and scientific collaborative work. In his spare time, Swapneel likes to read, watch movies, and travel to new places.

Vishal K. Singh did his B. Technology in Electronics and Communications Engineering from National Institute of Technology, Warangal in 2001. He worked at Cisco Systems, Bangalore from 2001 to 2004 before joining Columbia University as an M.S. student, and completed his M.S. in 2006. After his M.S. he worked at Motorola and NEC-Labs. Vishal started as a Ph.D. student in Fall 2008. His main research interests are system management, automated diagnosis in networks, IP QoS management, presence, and SIP. He is currently working with Prof. Schulzrinne.

Breannan Smith graduated from the University of North Carolina at Chapel Hill in 2007 with a B.S. in Computer Science and minors in Mathematics and Physics. As an undergraduate, Breannan's research focused on the simulation of nonlinear, viscoelastic fluids. Breannan is currently working with Prof. Eitan Grinspun within the Columbia Computer Graphics Group where he is investigating robust collision and contact schemes for the simulation of highly deformable systems with large degrees of multiple collisions and complex contact geometries.

Sampada Sonalkar graduated from Mumbai University, India with a B.E. in Computer Engineering in June 2001. She worked as a Research Assistant at Indian Institute of Technology Bombay and subsequently continued there for a M.S. in Computer Science from 2002-2004. She worked at General Motors Research in India from 2004-2006. Sampada did an M.S. in Computer Science at Columbia University from 2006-2007, and started as a Ph.D. student in Spring 2008. Her advisor is Prof. Luca Carloni. Sampada's main research interests are design automation and formal methods. She is currently working on system-level design of distributed embedded systems.

Li-Yang Tan graduated from Washington University in St. Louis with a B.A. in Mathematics and an M.Sc. in Computer Science in May 2006. He served in the Singapore Armed Forces from 2006 to 2008, and started as a Ph.D. student at Columbia in Fall 2008. Li-Yang's main research interest is theoretical computer science, and he is currently working with Professors Tal Malkin and Rocco Servedio.

Vishwanath Venkatesan graduated from Anna University, Chennai, India with a B.E. in Computer Science and Engineering in June 2006. He did his M.S. in Computer Science at University of Houston from 2006-2008, and started as a Ph.D. student at Columbia in Fall 2008. Vishwanath's main research interests are high performance architectures and parallel computing. He is currently working with Prof. Simha Sethumadhavan's group on automatic legacy code parallelization.

Jingyue Wu graduated from Tsinghua University with a B.E. in Computer Science in July 2008. He started as a Ph.D. student at Columbia in Fall 2008. Jingyue's main research interests are software systems and operating systems. He is currently working with Prof. Junfeng Yang and studying fixing errors of software systems.

Young Jin Yoon graduated from Korea University with a B.S. in Computer Science in June 2006. He did an M.S. in computer science at Columbia from 2006 to 2007, and started as a Ph.D. student in Spring 2008. Young Jin's main research interests are Network-on-Chip (NoC) and Computer Architecture. He is currently working with Prof. Carloni's group.

John R. Zhang graduated from the University of Calgary with a double B.S. in Computer Science and Mathematics in June 2008. He started as a Ph.D. student in Fall 2008, pursuing research in computer vision. John is currently working with Prof. John Kender studying gesture recognition.

Finally, one departing face deserves special mention. **Awilda Fosse**, Assistant to Prof. Joseph F. Traub and Editorial Coordinator, Journal of Complexity, retired in July after 8+ years in the CS Department. We will miss her professionalism, her positive attitude, and her great sense of humor.

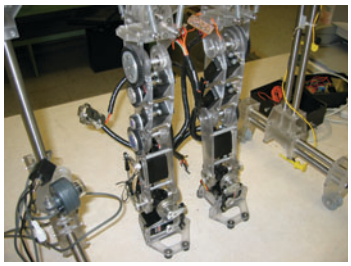
"What I did with my **summer** vacation"

After a busy year in classrooms and labs at Columbia, CUCS students branched out to a wide range of interesting jobs and internships over the summer. Here are brief snapshots of what a few computer science students did with their summers away from campus.

M.S. student **Rohit Gupta** writes, "This summer I worked as a Software Engineer Intern at Electronic Arts (EA) in Redwood City, CA. The most exciting part of my internship was the opportunity to work on 'The Sims 3,' the sequel to the highest-selling PC game in the history of video games (to date 'The Sims' has sold more than 100 million copies all over the world). Working on something so big, with such a big fan following, was like a dream come true, and being in a company where everybody is either young or young at heart was a totally different experience. Being a technical person, I appreciated the opportunity to interact with people having different backgrounds such as artists, producers, and directors—one of my team members had worked on movies like 'The Matrix' and 'Sin City.' 'The Sims 3' will be released in February 2009; you'd better grab a copy!"

Ph.D. student **Agustin Gravano** spent his summer as an intern at Google Research in New York. Agustin worked with Martin Jansche, a member of the Speech Processing Group, on post-processing the output of automatic speech recognition systems. Such systems produce raw word streams that, even with no recognition errors, are often difficult to read—not only for humans, but also for natural language processing tools, which usually expect formatted text as input. During his internship Agustin helped develop machine learning tools for automatically inserting punctuation symbols, capitalizing words, and formatting numbers. It was also a great opportunity to meet colleagues from the industry, and to experience the famous working conditions at Google, with endless perks and an amazing work environment.

Ph.D. student **Homin K. Lee** spent his summer vacation as an intern at the IBM Almaden Research Center in San Jose, CA. He worked in the Theory Group under the guidance of Vitaly Feldman and T. S. Jayram. Homin is happy to report that the weather never changed for all three months he was there.



Undergraduate student **Benjamin Mann** (SEAS '11) writes: "Last summer, I was accepted for NASA's 10 week Robotics Academy. With two other undergraduates and a masters student, I worked for PI Mehran Armand on the stereo vision system for a bipedal robot. The vision system will be used for many robotic systems as a testing platform for obstacle avoidance, path planning, and terrain handling algorithms, and more. Outside of work, the 26 members of the Robotics Academy heard lectures from robotics and space industry professionals twice a week. We traveled to MIT, Carnegie Mellon, and Johnson Spaceflight Center to learn about their robotics programs. At Carnegie Mellon, we rode in the DARPA Urban Challenge winning vehicle, Boss, in autonomous mode on a closed course."

SEAS undergraduate Tristan Naumann (second from right) had a productive summer at Google NYC.

Undergraduate student **Tristan Naumann** (SEAS '09) writes: "This summer I worked at Google NYC as an Associate Product Management (APM) Intern on Google Print Ads. My internship required a unique combination of broad technical knowledge, communication and organization in order to create new features and manage their development. The people I met—other interns, full-timers, and even founders—had diverse backgrounds but were all extraordinarily gifted at whatever their jobs entailed. On top of it all, everyone was full of energy and had a great sense of humor, making this one of the best summers I've ever had! I even made it onto the official Google blog for the success of our cube decoration project *(see image above)*."

Ph.D. student **Snehit Prabhu** spent an eventful summer in sunny Boston, at the Broad Institute of MIT and Harvard. When he wasn't biking through the quaint, sunburnt-brick lined streets around Quincy Market, he was going on long hikes or learning Lindy-Hop. In his free time, he studied the ways in which populations of Baker's Yeast grown under harsh environments (like high salt and food scarcity) responded to the evolutionary pressures imposed on them. With each successive generation, the population accumulated mutations in its DNA, in the hope of having fitter, more robust progeny. Comparing the DNA sequences of populations grown separately under these trying circumstances, Snehit's team found statistically significant differences in genetic constitution between them. Identifying biological functions that were altered as a result of this DNA variation provided insights into natural selection and the mechanisms by which these organisms evolve over time.

Ph.D. student **Andrew Wan** writes, "This summer was filled with adventure. First, I made a visit to Tsinghua University in Beijing, where I gave a talk on recent work with Dachman-Soled, Lee, Malkin, Servedio and Wee. Afterwards, I traveled around China with my father. There is a lot of construction in China! I spent the rest of the summer at SRI in Palo Alto working with Chris Peikert on trying to break the NTRU signature scheme. I was glad to pay my family in South San Francisco a visit. Finally, I also made a trip to Reykjavik to attend the ICALP conference."

Undergraduate **Anthony Yim** (SEAS '10) spent his summer working at a technology start-up company called Peek. Peek is currently creating a mass-market, super consumer friendly and contract free mobile email device to be launched in mid Sept 2008. Anthony worked with **Amol Sarva** (CC '98), a Columbia alumni who co-founded Virgin Mobile and was an original member of the Wireless Founders Coalition for Innovation. While at Peek, Anthony prepared the device for the nation-wide launch by performing extensive testing and readying Peek's CRM system. Anthony also performed and presented independent research on mobile operating systems and processors for the company. In the process, he has learnt a lot about the workings of the telecommunications industry. While not at Peek, Anthony spent his time hiking on the Appalachian Trail and exploring New York City.

The Department at Play: **Spring Barbecue**

Barbecues in the courtyard are a cherished tradition in the Columbia Computer Science Department. Here are some photos from the most recent event.

Ph.D. student **Salman Baset** received a Marconi Young Scholar award for 2008. The award recognizes the accomplishments and promise of Ph.D. students in their early years who are working in the areas of communications and networking. Baset's Ph.D. work is on designing algorithms and protocols for peer-to-peer voice and video systems. The work leverages the distributed nature of peer-to-peer systems to solve connectivity issues, making routing and remote access service easier.

Computer Science Ph.D. students **Oliver Cossairt** and **Alexander Gusev** were awarded prestigious NSF graduate fellowships.

Olivier Cossairt, advised by Prof. **Shree Nayar**, has been selected for the NSF Graduate Fellowship to further his work on intelligent displays. Intelligent displays are new types of visualization systems that sense and react to their physical environment. Using these displays, real and digital objects are indistinguishable in appearance, enabling new possibilities for rich user interaction in fields as diverse as medical imaging, military visualization, and entertainment.

Alexander Gusev, advised by Prof. **Itsik Pe'er**, has been awarded an NSF Graduate Fellowship for his research work in computational genetics. Genetic evidence shows that many pairs of individuals purported as unrelated to one another actually share an ancestor within the last few generations. Unfortunately, the computational barrier of comparing all pairs to one another prevented such analysis on a large scale. Sasha developed a linear time method for such all-against-all comparison, becoming the first to analyze ancestry of thousands of individuals, and finding surprising results with implications to population genetics and disease research.

Computer Science Ph.D. students **Melinda Agyekum** and **Ryan Overbeck** were awarded Intel Fellowships. These highly competitive two-year fellowships include a cash award, a laptop, an Intel mentor, and an internship.

Melinda Agyekum, advised by Prof. **Steven Nowick**, was selected for her work in asynchronous digital systems. Asynchronous digital circuits perform synchronization and communication without using a global clock, and thereby can provide greater flexibility and timing-robustness in handling on-chip and off-chip communication. The goal of Melinda's work is to provide low-power encoding techniques that will allow asynchronous communication to become more tolerant of dynamic variability (e.g., soft-errors, cross-talk, noise, etc.) which has become an increasing problem due to device scaling.

Ryan Overbeck, advised by Prof. **Ravi Ramamoorthi**, was selected for his research on real-time ray tracing on the CPU. Ray tracing is the core of many physically based algorithms for rendering 3D scenes with global illumination (shadows, reflections, refractions, indirect illumination, and other effects), but hasn't achieved interactive rendering times on commodity computers until recently. Ryan develops algorithms to ray trace 3D scenes with high quality shadows, reflections, and refractions providing a higher degree of realism to interactive content.

Computer Science Ph.D. student **Carlos-Rene Perez** was awarded a six-year fellowship from the National Physical Science Consortium (NPSC). Carlos-Rene is advised by Professors **Angelos Keromytis** and **Jason Nieh**, and the fellowship is sponsored by the NSA. The primary objective of the NPSC is to increase the number of qualified U.S.–citizen Ph.D.s in the physical sciences and related engineering fields, emphasizing recruitment of a diverse applicant pool of women and historically underrepresented minorities. NPSC accomplishes this objective by assisting corporations and government agencies and laboratories in awarding doctoral fellowships to outstanding U.S. students.

Computer Science Ph.D. students **Paul Blaer** and **Matei Ciocarle** were featured in an article in La Nouvelle Republique on their work on scanning the cathedral of Bourges.

Professors **Peter Belhumeur** and **Shree Nayar** are participating in an NSF MURI project on human recognition. A research team led by Prof. **Rama Chellappa** (ECE/UMIACS/CS) has won a 2008 MURI award for their proposal to develop face, gait, long-distance speech and other motion-based human recognition algorithms tailored to the maritime domain. The MURI project will be coordinated by a team of researchers from University of Maryland with partnering institutions in Columbia University, University of California at Colorado Springs and University of California at San Diego, and international researchers from University of Southampton, UK and University of Queensland, Australia. The team will design and develop novel sensors, algorithms and systems for maritime biometrics.

Professor **Luca Carloni**, together with Professor **Keren Bergman** (Electrical Engineering), received a grant from the National Science Foundation entitled "Photonic Interconnection Networks for Chip-Multiprocessor Computing Systems" in the CPA-CSA program. This is a competitive program in which, this year, only about 10-15% of the proposals received funding. This research project aims to harness the recent extraordinary advances in nanoscale silicon photonic technologies for developing optical interconnection networks that address the critical bandwidth and power challenges of future chip multiprocessor-based systems. The PIs will investigate the complete cohesive design of an on-chip optical interconnection network that employs nanoscale CMOS photonic devices and enables seamless off-chip communications to other CMP computing nodes and to external memory. System-wide optical interconnection network architectures will be specifically studied in the context of stream processing models of computation.

Professor **Luis Gravano** and Ph.D. student **Hila Becker** received a Google Research Award to work on local event search. Their project focuses on two problems associated with local event search, namely, how to identify events of all sizes—including small, not-so-prominent events not necessarily covered in mainstream sources—and how to determine the "geographical scope" of events—beyond their explicit location. The project will use the wealth and variety of sources that are available over the Web to identify and characterize events, in turn to produce expressive, high-quality local event search results.

Professor **Luis Gravano** received a three-year National Science Foundation grant for the project titled "Beyond Keyword Search: Enabling Diverse Structured Query Paradigms over Text Databases", part of the NSF Information and Intelligent Systems: Advancing Human-Centered Computing, Information Integration and Informatics, and Robust Intelligence program. Keyword search is the prevalent query paradigm for text, but is often insufficiently expressive for complex information needs that require structured data embedded in text. To move beyond keyword search, this project exploits information extraction technology, which identifies structured data in text, to enable structured querying. The project explores a wealth of structured query paradigms and develops specialized cost-based query optimizers for each query paradigm, accounting for the efficiency and, critically, the result quality of the query execution plans. The project will also provide data sets and source code, for experimentation and evaluation, to the community at large over the Web, at *http://extraction.cs.columbia.edu/*.

Professor **Julia Hirschberg** was named as one of the 12 initial Fellows of the International Speech Communication Association (ISCA). The purpose of ISCA is to promote, in an international world-wide context, activities and exchanges in all fields related to speech communication science and technology. The association is aimed at all persons and institutions interested in fundamental research and technological development that aims at describing, explaining and reproducing the various aspects of human communication by speech. ISCA has about 1500 members. Prof. Hirschberg has also served as past president of ISCA.

Professor **Julia Hirschberg**, along with Professor **Ani Nenkova** at the University of Pennsylvania, have received a grant from the National Science Foundation titled "Corpus-Based Studies of Lexical, Acoustic-Prosodic, and Discourse Entrainment in Spoken Dialogue". Participants in human-human conversation often entrain to one another, adopting the vocabulary and other behaviors of their partners. The project will investigate many types of entrainment in two large corpora of human-human conversations to improve system behavior in Spoken Dialogue Systems (SDS). Prof. Hirschberg and Nenkova want to discover which types of entrainment occur generally across speakers and which seem to be speaker-specific, which types of entrainment can be reliably linked to task success and perceived naturalness, and which types of entrainment can be automatically modeled in SDS. The project focuses on determining which types of system entrainment to users will be most important to users and most feasible for SDS. The team will also provide publicly available annotated corpora for future research by others. Prof. Nenkova received her doctorate from Columbia University in 2006.

Professor **Angelos Keromytis** was appointed to the scientific advisory board of CERTH (Center for Research and Technology, Hellas) by the Greek ministry of development. The Centre for Research and Technology Hellas (CE.R.T.H.), the largest research centre in Northern Greece, was founded in March 2000. CERTH is a non-profit organization that directly reports to the General Secretariat for Research and Technology (GSRT), of the Greek Ministry of Development. The mission of CERTH is to carry out fundamental and applied research with emphasis on development of novel products and services of industrial, economic and social importance in a range of different fields.

Professor **Angelos Keromytis** received a grant from Intel: "Tracking Sensitive Information Flows in Modern Enterprises". The project will investigate mechanisms for implementing information accountability and visualization in large-scale distributed enterprise environments. Specifically, Angelos will investigate the use of Virtual Machine Monitors (VMMs) and distributed coordination protocols to efficiently track the flow of sensitive information (or, more generally, information of interest to the administrator) throughout a distributed system. Although much work has been done on VMMs in recent years, the focus has been on more efficient resource utilization and (from a security standpoint) component isolation; little to no work has been done on fine-grained information flow tracking within a single system and across system boundaries.

The National Science Foundation is funding a new CyberTrust project by Professor **Tal Malkin** and her colleague at the University of Connecticut titled "Tamper Proofing Cryptographic Operations". This research project focuses on the development of cryptographic mathematical models and constructions that address realistic security requirements at the implementation level. This is a fundamental problem as cryptographic security formalisms are often criticized for lack of relevance given the wide range of attacks available at the implementation level. The research extends models of cryptographic attacks to include various forms of private data tampering and access and brings the theory of cryptographic constructions closer to security concerns in practice. In particular, the tamper proofing of a wide set of cryptographic primitives is considered in an extended adversarial setting, such as digital signatures, public key encryption, secure function evaluation, as well as arbitrary cryptographic functions. This research thus explores the boundaries of what is achievable algorithmically and practically through cryptographic means.

Professor **Steve Nowick** received a grant from the National Science Foundation, along with Professor **Uzi Vishkin** of the University of Maryland, to design new tools that allow to design multi-core systems that combine synchronous and asynchronous elements. The grant is part of the Computing Processes and Artifacts (CPA) program; only about 10-15% of the proposals in the "Design Automation for Micro and Nano Systems" topical area were funded. The proposal focuses on a particular existing easy-to-program and easy-to-teach multi-core architecture. It then identifies the interconnection network, connecting multiple cores and memories, as the critical bottleneck to achieving lower overall power consumption. The target is to substantially improve the power, robustness and scalability of the system by designing and fabricating a high-speed asynchronous communication mesh. The outcome of the work could make a step in the paradigm shift from serial to parallel that the field is now undergoing; the resulting first-of-its-kind partly-asynchronous high-end massively-parallel on-chip computer could push the level of scalability beyond what is currently possible and have a broad impact in supporting parallel applications in much of computer science and engineering.

The National Science Foundation is funding a joint GATech, Bell Labs and Columbia next-generation Internet project. The project, titled "NetSerV–Architecture of a Service-Virtualized Internet," investigates how to allow network nodes to offer services beyond just routing packets. The Columbia team is led by Professor **Henning Schulzrinne**. To address the limitations of services in the current Internet, this work presents a clean slate Internet architecture, called NetServ, based on the concepts of service virtualization. NetServ strives to break up the functions provided by Internet services and to make these functionalities available as modular building blocks for network services. This project addresses five major research challenges in the new service architecture: **i)** the definition of requirements for a service-virtualized Internet architecture, **ii)** the design of an architectural framework for modular, virtualized services, **iii)** the identification of an initial set of key building blocks, which together can provide a foundation for common network services, **iv)** the development of mechanisms and protocols for service discovery and service distribution, and **v)** the design and implementation of a content distribution service based on the NetServ architecture. The feasibility of the NetServ approach will be demonstrated by building the prototype of a content distribution service. Overall, the objective of this work is to develop an architecture that provides an efficient and extensible architecture for core network services, to implement a prototype of the new architecture, design and implement a prototype of a content distribution service to

demonstrate the feasibility of NetServ, and to evaluate the new architecture in a simulation environment as well as on a GENI-like test bed.

The National Science Foundation is supporting a joint project on next-generation emergency calling services, with researchers from the University of North Texas, Columbia University, and Texas A&M University. The Columbia University portion of the project is led by Professor **Henning Schulzrinne**. Traditional 9-1-1 systems, which date back to the 1970s, support only voice, while non-emergency communications now feature other media. Adding additional media for 9-1-1 presents opportunities and challenges. The project will develop a testbed that will enable research on understanding and analysis of next generation 9-1-1 services. This is particularly important as both state and federal governments are in the process of planning next-generation emergency communication platforms, unfortunately often without adequate vendor-neutral testing and evaluation. The project plans to investigate issues related to locating 9-1-1 callers, securing Public Safety Answering Points, ensuring continuous availability of 9-1-1 services during large-scale emergencies, predicting emergencies, providing citizen alerts ("reverse 9-1-1"), improving inter-agency coordination and enhancing 9-1-1 services for the deaf and hearing-impaired using video phones and instant messaging. The research results will translate into engineering guidelines and be disseminated across government organizations, standards bodies such as IETF and National Emergency Number Association (NENA) and 9-1-1 centers.

Professor **Henning Schulzrinne** was appointed to the Internet2 Applications, Middleware & Services Advisory Council (AMSAC). The Applications, Middleware, and Services Advisory Council (AMSAC) is responsible for advising the Board and management on matters relating to the support and adoption of applications, middleware, security and other capabilities across the Internet2 membership and its collaborators around the globe. The AMSAC is responsible for interacting directly with other key advisory committees on technical and service issues. It will also provide advice on Internet2's efforts to support applications and middleware for teaching and learning as well as for research, and for advice on the investment of resources for current and future initiatives.

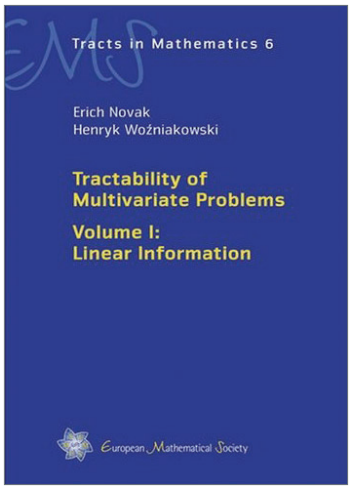
Professors **Simha Sethumadhavan** and **Sal Stolfo**, together with Dr. **Michael Locasto** (Dartmouth, former CUCS Ph.D.) and Professor **David August** (Princeton), have received funding from the Defense Advanced Research Projects Agency (DARPA) to study techniques for parallelizing legacy software codes for multi-core processors. This research effort will leverage recent discoveries of latent parallelism in sequential codes and improvements in machine learning to create a new automatic parallelization system. The parallelization system may offer performance improvements at historical rates for legacy software (on multi-cores).

Ph.D. student **Charles Shen** received a Best Student Paper award at IPTComm 2008, held in Heidelberg, Germany. The paper "SIP Server Overload Control: Design and Evaluation" was co-authored by Charles Shen and Prof. **Henning Schulzrinne**. It describes and evaluates mechanisms so that VoIP servers can continue to operate at full capacity even under severe overload. Such overload may occur during natural disasters or mass call-in events, such as voting for TV game show contestants. Without these measures, servers are likely to suffer from congestion collapse.

Joseph F. Traub, Edwin Howard Armstrong Professor of Computer Science, has been elected to the Board of Directors of the Marconi Society. The Society is best known for the Marconi Prize, considered the most prestigious award in the field of communications and the Internet. Among its other activities, the Society holds regular forums on topics of societal importance.

Professors **Henryk Wozniakowski** and **Joseph Traub** have received an NSF grant for "Quantum and Classical Complexity of Continuous Problems". The investigators are studying the following general question: If physicists and chemists succeed in building quantum computers, which continuous problems arising in science and engineering can be solved much faster on a quantum computer than on a classical computer? Examples of continuous problems are path integration, the Schrodinger equation, high-dimensional approximation, continuous optimization, and integral equations. To obtain the power of quantum computation for continuous problems one must know the computational complexity of these problems on a classical computer. This is exactly what the investigators have studied for decades in the field of information-based complexity. Some specific goals of the project are to study the power of randomized quantum queries; to investigate whether the randomized classical setting suffers the "curse of dimensionality" for the problem of computing the ground state energy of a system; and to study algorithms and initiate the study of the computational complexity of the Schrodinger equation in the worst case and randomized settings on a classical computer and in the quantum setting.

Professor **Henryk Wozniakowski** was inducted into the Polish Academy of Sciences and also received an honorary doctorate from the Friedrich-Schiller-University in Jena, Germany. The Polish Academy of Sciences *(http://www.pan.pl/english/)* is a state scientific institution founded in 1952. It functions as a learned society acting through an elected corporation of top scholars and research organizations, via its numerous scientific establishments. It has also become a major scientific advisory body through its scientific committees. The honorary doctorate (Dr. rer. nat. hc) cited Prof. Wozniakowski's foundational contribution to numerical methods, particularly the deep insights due to the new discipline of information-based complexity and the work on the "curse of dimensionality" that helps determine which high-dimension problems are solvable. The Friedrich-Schiller University in Jena was founded in 1588.



The first volume of Professor **Henryk Wozniakowski**'s book, Tractability of Multivariate Problems, written jointly with **Erich Novak** (University of Jena, Germany) has been published by the European Mathematical Society.

Recent alumni of the DNA (Distributed Network Analysis) lab have pursued a range of interesting career paths after graduation.

- **Hanhua Feng** (Ph.D. '07) is working for IBM Research Watson as a member of the technical staff.
- **Abhinav Kamra** (Ph.D. '07) is working for Citigroup as a financial analyst.
- **Raj Kumar** (Ph.D. '08) is working for Telcordia as a member of the technical staff.
- **Patrick Lee** (Ph.D. '08) is doing a 1-year postdoc at University of Massachusetts, and will start a tenure-track faculty job in the Computer Science Department of the Chinese University of Hong Kong (CUHK) in 2009.

Issy Ben-Shaul (Ph.D. '95) got tenure from the Technion, then left to found a startup, Actona, which he sold to Cisco in 2004. After working for Cisco for a few years, he has recently founded a new startup, Metis Technologies, where he is currently the CTO.

Aaron Davis (SEAS '02, Computer Engineering) recently joined Tudor, a hedge fund headquartered in Greenwich. He is now working in their office in Singapore as a programmer and trader.

Kyle Dent (M.S. '07) writes, "I recently took a new position with the Palo Alto Research Center (formerly Xerox PARC) as a Senior Member of the Research Staff. Previously I had worked at HP for about 12 years. My Columbia masters degree was certainly a factor in obtaining the new position."

George Heineman (Ph.D. '96) writes, "I am an associate professor in Computer Science at Worcester Polytechnic Institute (WPI) in Worcester, MA, where I've been since I received my Ph.D. I'm happy to announce the publication of Algorithms in a Nutshell by O'Reilly Media, Inc. The web page for the book is: *http://oreilly.com/catalog/9780596516246*. This book serves two audiences. For undergraduate students, this reference is a concise CliffsNotes-like™ introduction to the most commonly used algorithms in the Computer Science Literature. For the working professional, the book contains fully-coded solutions to show how the algorithms can rapidly be coded within your own software systems to achieve improved code performance."

Dr. **Arnold Kim** (B.S. '96) was featured in a New York Times Technology section article "My Son, the Blogger: An M.D. Trades Medicine for Apple Rumors." Dr. Kim has been blogging about Apple since 2000; his blog *MacRumors.com* is one of the most popular technology Web sites, attracting more than 4.4 million people and 40 million page views per month.

Jim Kurose (Ph.D. '84) has been appointed Interim Dean of the College of Natural Science and Mathematics (the college home of Computer Science Department) at the University of Massachusetts. He was also recently elected to the Board of Directors of the Computing Research Association (CRA). Jim joined the UMass in 1984 after finishing his Ph.D. in Computer Science from Columbia in networking under the supervision of **Mischa Schwartz** and **Yechiam Yemini**. Two of his former Ph.D. students (**Henning Schulzrinne** and **Dan Rubenstein**) and a former postdoctoral researcher (**Vishal Misra**) are on the Columbia Computer Science faculty.

Michael Locasto (Ph.D. '07) writes, "After my postdoc at the Institute for Security Technology Studies (ISTS) at Dartmouth College working with Sean Smith and David Kotz, I've joined the faculty at George Mason University as a Visiting Research Assistant Professor. My position is funded by an I3P Fellowship that I applied for while at Dartmouth. At Dartmouth, I was involved in organizing and running the SISMAT program, which is an educational outreach program to help foster information security expertise at 4-year undergraduate institutions that do not have local systems or security curriculum or expertise. A news release from Dartmouth is here: *www.dartmouth.edu/~news/releases/2008/06/23.html*."

Ajit Singh (Ph.D. '90) was appointed as the Chief Executive Officer of BioImagene. BioImagene is a leading provider of innovative and affordable digital pathology solutions for cancer diagnosis and pre-clinical research.

Dragomir Radev (Ph.D. '99), an associate professor of computer science, information, and linguistics at the University of Michigan, was the head coach for the U.S. team which competed at the Sixth International Linguistics Olympiad in Slanchev Bryag, Bulgaria. The U.S. team was selected from over 750 high school students who participated in qualifying events held at 77 sites around the U.S. and Canada this past winter. The team captured 11 out of 33 awards, including gold medals in individual and team events. This year's Olympiad featured 16 teams from around the world, including Bulgaria, Estonia, Germany, Latvia, the Netherlands, Poland, Russia, Sweden, South Korea and Slovenia. Each problem presented clues about the sounds, words or grammar of a language the students had never studied, such as Micmac, a Native American language spoken in Canada, the New Caledonia languages of Drehu and Cemuhi, as well as several historical Chinese dialects. They were then judged by how accurately and quickly they could untangle the clues to figure out the rules and structures of the languages to solve the problem.

Sam Somech (M.S. '80) was named Chairman of the Board of Directors of GigaSpaces Technologies, a leading provider of a scale-out application server for Java, .Net and C++ environments.

Krish Sridhar (M.S. '03) writes, "After working at Morgan Stanley for the last 5+ years, I decided to go back to school and get an MBA with a future interest in Venture Capital or Strategy. I just moved to London a few weeks back and am enrolled in the program at London Business School *(www.london.edu)*. The classes are just getting started and we have students from over 60 countries here making it an incredibly diverse and exciting programme."

Andy Wolfe (Columbia College '07) writes, "I am leaving my position as a software developer with IBM. After one short year, a great opportunity has presented itself here in Austin. I am joining the Product Management team at Bazaarvoice, *http://www.bazaarvoice.com*, next week. Bazaarvoice is a mature start-up in the social commerce space."

Stay in Touch!

Visit the CUCS Alumni Portal at **https://mice.cs.columbia.edu/alum** to:

- Update your contact information
- Look at recent job postings
- Get departmental news

If you know another alumni who may not be receiving the newsletter, please forward the Alumni Portal link to them.

In the spirit of environmental responsibility, we are moving towards more electronic (and less hard-copy) distribution of the newsletter. If you'd like to continue receiving paper copies of the newsletter, please visit the Alumni Portal to sign up.

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**

Read the CUCS Newsletter online at **http://www.cs.columbia.edu/resources/newsletters**

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