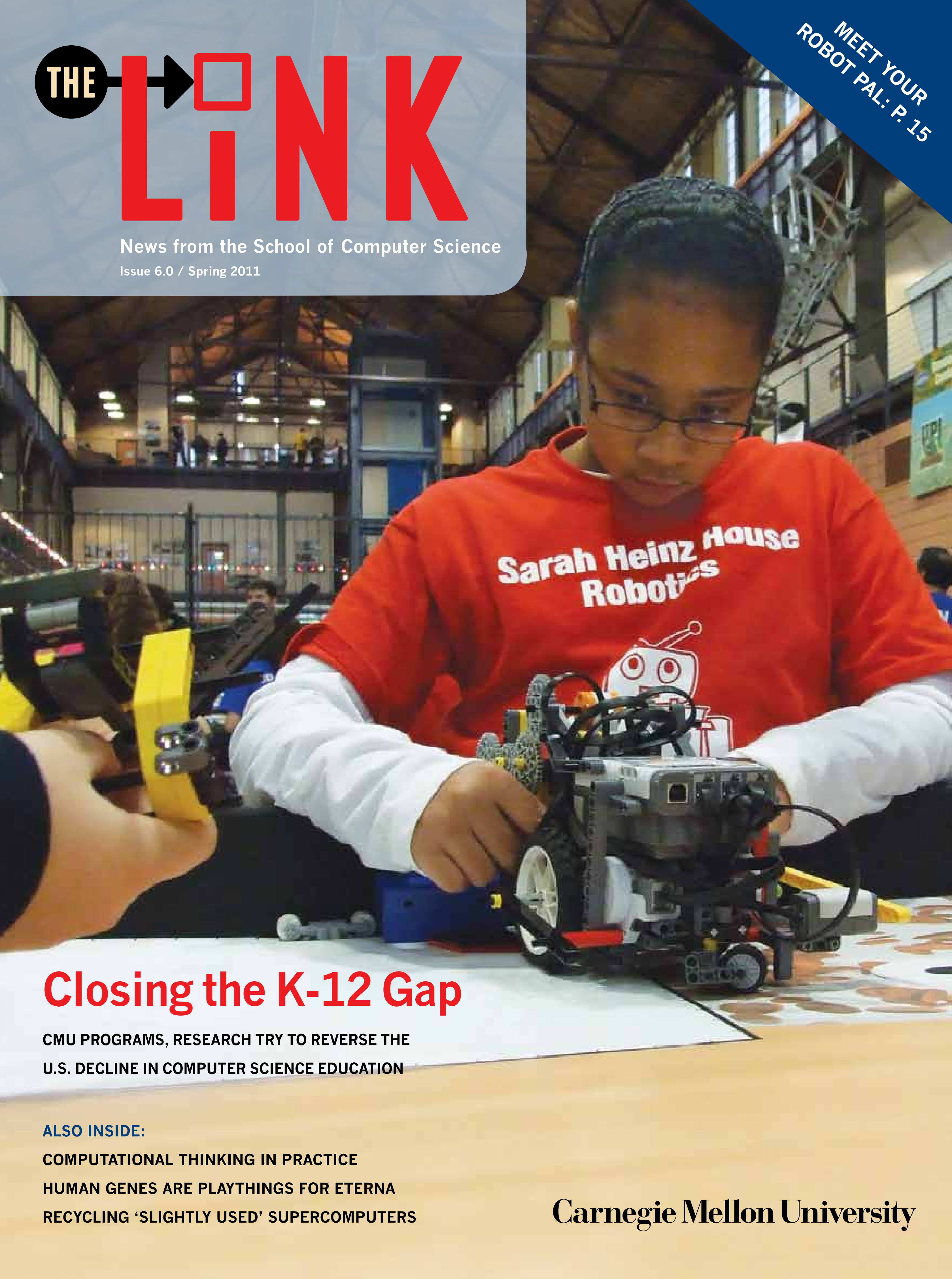
THE LINK

News from the School of Computer Science

Issue 6.0 / Spring 2011

MEET YOUR ROBOT PAL: P. 15



Closing the K-12 Gap

CMU PROGRAMS, RESEARCH TRY TO REVERSE THE U.S. DECLINE IN COMPUTER SCIENCE EDUCATION

ALSO INSIDE:

COMPUTATIONAL THINKING IN PRACTICE

HUMAN GENES ARE PLAYTHINGS FOR ETERNA

RECYCLING 'SLIGHTLY USED' SUPERCOMPUTERS

Carnegie Mellon University

THE LINK

The Link provides a mosaic of the School of Computer Science: presenting issues, analyzing problems, offering occasional answers, giving exposure to faculty, students, researchers, staff and interdisciplinary partners. The Link strives to encourage better understanding of, and involvement in, the computer science community.

Editor-in-Chief
Randal E. Bryant

Managing Editor
Jason Togyer

Contributing Writers
Jennifer Bails, Tina Carr, Ken Chiacchia, Mark Dorgan, Meghan Holohan, Tom Imerito, Byron Spice

Photography
John Barna, Stephen M. Chabassol, Catherine Copetas, Tim Kaulen, Wade H. Massie, Brad A. Morris, Paul A. Selvaggio

Graphic Design
Melissa Stoebe
Communications Design & Photography Group

Office of the Dean
Gates Center 5113
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Randal E. Bryant, dean
Tina Carr (HNZ'02), director of alumni relations
Philip L. Lehman (CS'78,'84), associate dean for strategic initiatives
Byron Spice, director of public relations
Jason Togyer (HS'96), writer/editor

Phone: 412-268-8721
Email: TheLink@cs.cmu.edu
Web: link.cs.cmu.edu
Facebook: facebook.com/SCSatCMU
Twitter: twitter.com/SCSatCMU

Produced for the School of Computer Science by the Communications Design & Photography Group, February 2011, 11-399.

CALENDAR OF EVENTS

All events to be held at the Carnegie Mellon University campus in Pittsburgh, unless otherwise noted. Dates and locations are subject to change without notice. Visit calendar.cs.cmu.edu for a complete and current listing of events.

March 3
CNBC Colloquium:
David Poeppel, New York University
4 p.m., Biomedical Science Tower 3,
Room 6014, University of Pittsburgh

March 4
Alumni Network Night D.C.
Washington, D.C.

Mid-Semester Break: No classes

March 4–6
OurCS 2011: Workshop for Undergraduate
Women in Computer Science
Gates & Hillman Centers
www.cs.cmu.edu/ourcs

March 7–11
Spring Break: No classes

March 8
Alumni Network Night Boston
Hosted by Zipcar

March 18
2011 Smiley Award Ceremony and Reception
Location TBA

March 21
The Dickson Prize in Science
Award Presentation
David A. Tirrell,
California Institute of Technology:
"Reinterpreting the Genetic Code"
4:30 p.m., Mellon Institute Auditorium

March 29
CNBC Colloquium:
Michael Tomasello,
University of Leipzig and
Manchester University
4 p.m., Western Psychiatric Institute
and Clinic, Second Floor Auditorium,
University of Pittsburgh

April 6
CSD Faculty Meeting
3:30 p.m., Gates Center 6115

April 8
SCS Graduate Student
Appreciation Day
4–6 p.m., Perlis Atrium,
Newell-Simon Hall

April 10–11
Spring Sleeping Bag Weekend

April 14–16
Spring Carnival and Reunion Weekend
No classes

April 16
SCS and ECE Alumni Spring Carnival
Reception
1–3 p.m., Gates Center 6115

April 29
Last day of classes, spring semester

May 2–10
Final exams

May 4
CSD Faculty Meeting
3:30 p.m., Gates Center 6115

May 15
Commencement

May 16
Summer classes begin

May 30
Memorial Day: No classes

June 1
CSD Faculty Meeting
3:30 p.m., Gates Center 6115

July 4
Independence Day: No classes

Aug. 29
Fall semester begins

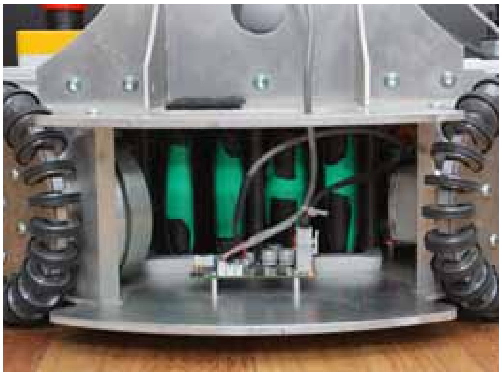
Sept. 5
Labor Day: No classes

C O N T E N T S

10 / Cover Story: Closing the Educational Gap

A new report co-authored at CMU concludes that budget pressures and federal mandates created by the No Child Left Behind Act have caused U.S. elementary and high schools to cut or drop computer science education for grades K–12. That means fewer American students are pursuing computer science and engineering careers. A variety of programs developed at the School of Computer Science is trying to reverse the trend.

By Meghan Holohan



15 / Feature: Making a Mechanical Pal

A TV-watching child of the 1960s or '70s would be forgiven for assuming she'd have a robot pal by now. But developing practical social robots turned out to be a lot harder than futurists suspected two generations ago. Several new social robots are expected to begin limited testing at SCS this year, and their developers say useful robotic companions need more than just happy faces.

By Jason Togyer

Issue 6.0 / Spring 2011

On the Cover:

A member of the robotics team at Sarah Heinz House on Pittsburgh's North Side prepares its entry in the 12th annual FIRST Lego League tournament, sponsored by CMU's Robotics Academy. The tournament was held Dec. 4 at the National Robotics Engineering Center in the city's Lawrenceville neighborhood.

The Robotics Academy provides free software and other support materials that enable educators to incorporate computer science and engineering lessons into their existing curricula.

A new report co-authored at CMU warns that computer science education in U.S. schools is falling behind that of other developed nations. Programs from the School of Computer Science, such as the Robotics Academy, Andrew's Leap and CS4HS are helping reverse the decline. Our story begins on page 10.

(Photo by Brad A. Morris)

Randal E. Bryant

As we move into 2011, we see people in the School of Computer Science engaged in a broad range of educational and research topics. We see our role as computer science educators expanding to include service not just to our own students, but to all of society.

Locally, we have many initiatives to reach out to pre-college students, including our long-running Andrew's Leap summer program. In terms of our nation, the trends for high school education in computer science are going in the wrong direction, as Mark Stehlik and Leigh Ann Sudol pointed out in their recent report for the Association of Computing Machinery, "Running on Empty." This issue's cover story, beginning on page 10, provides an overview of some of our initiatives.

On the brighter side, Jeannette Wing's efforts to formulate the core principles of computer science, which she terms "computational thinking" ("Research Notebook," page 20) have catalyzed efforts worldwide to better understand the larger intellectual role of computer science.

In many ways, we can see our research projects having broad impact worldwide. Our social robots are starting to explore the roles envisioned by futurists and science fiction writers many years ago. Online gamers will be able to use their talents to help scientists explore the structure of different RNA molecules. Our systems researchers will be getting access to large-scale systems created by repurposing obsolete supercomputers. The Never-Ending Language Learning (NELL) project continues to learn interesting and important facts from the worldwide web. All in all, these are exciting times for CMU's School of Computer Science!

Randal E. Bryant
Dean and University Professor
School of Computer Science

FEEDBACK LOOP

Looking Back, Looking Ahead

In your Fall 2010 issue, I found quite a bit of material from my days (1968–1973) in the Computer Science Department at CMU. Gordon Bell was on my thesis commitee, and I took many courses from the other professors mentioned.

In addition, I was very interested in the material on the changes in the curriculum at SCS. I am currently teaching software engineering at the Computer Engineering Department of the School of Engineering and Architecture at Yeditepe University in Istanbul.

Thank you for a wonderful publication.

Birol Aygün (CS'73)
Yeditepe University, Istanbul, Turkey

Keep it Up!

We appreciate and thank you for your efforts in sharing news in SCS via The Link magazine. I am always pleased to learn about "what's happening" in our school, and am often intrigued by the various projects and work being done in other departments.

Congratulations on a job well done.

Ann Papuga
Senior Program Manager, Institute for Software Research

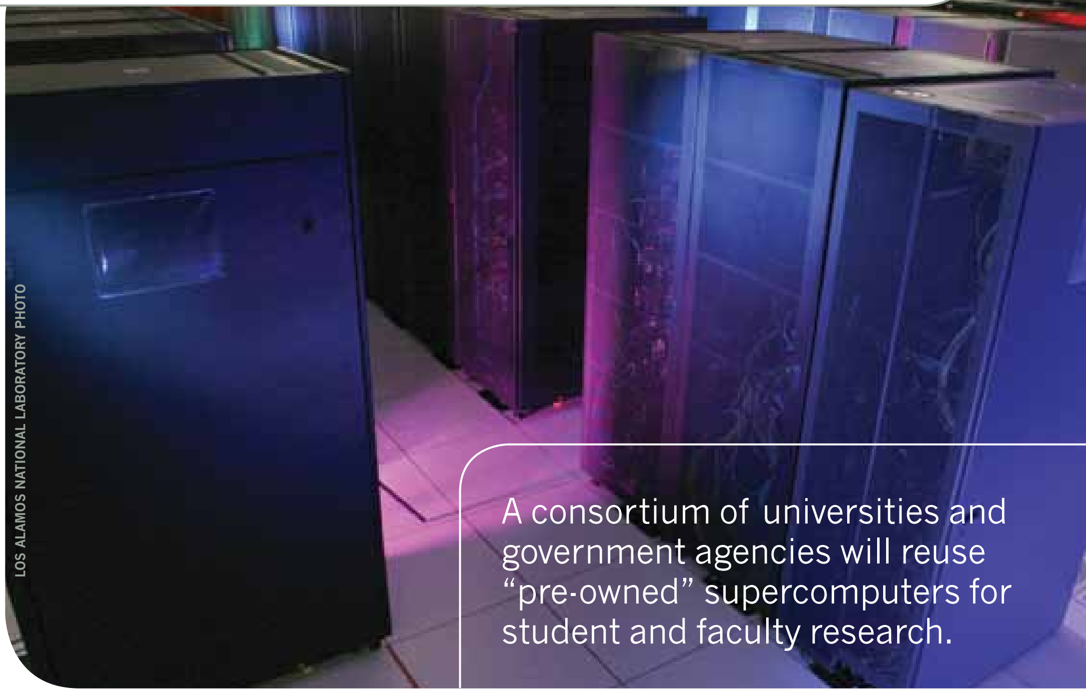
Citizen Science

Eric Paulos' piece on "Citizen Science" (Research Notebook, Spring 2010) was a very nice article about the potential of data collection by non-scientists. There is enormous precedent for this in amateur astronomy, weather monitoring, bird watching, botany and other fields. The scholarly literature has greatly benefitted from contributions and discoveries by amateur scientists.

Forrest M. Mims III
Columnist, San Antonio, Texas, Express-News
(via link.cs.cmu.edu)

Do you have a comment about a story you've read in The Link? A suggestion for a topic you'd like to see us cover? A correction or a compliment? Send your bouquets and brickbats to The Link Magazine, Office of the Dean, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, or email TheLink@cs.cmu.edu. Comments may also be posted to our website, link.cs.cmu.edu. Messages of all sorts are cheerfully accepted, though spam is fed to the dragon.

Super-Sized Recycling



High performance computing crossed the petascale threshold in 2008 when Los Alamos' RoadRunner, shown here, came online. Students and researchers will get unprecedented access to large-scale clusters thanks to a new effort to recycle obsolete supercomputers.

By Jennifer Bails

It's hard out there for a supercomputer.

As soon as you're up and running, you're put to work crunching terabytes of data for computational biologists, astrophysicists and all of the other pushy scientists who expect instant results.

You toil alone for years in a freezing cold room, and after that, do you get any show of appreciation? Yeah, right. They call you slow and obsolete, and you get tossed in a landfill to spend eternity with millions of ordinary cell phones, laptops and other e-waste. That's after they destroy your memory. And then they have the gall to replace you with a less experienced—albeit more computationally intensive and power efficient—machine.

At the U.S. Department of Energy's Los Alamos National Laboratory (LANL) in New Mexico, up to 5,000 machines from large-scale supercomputers are disposed of in this way each year, according to Gary Grider, deputy division leader of the lab's High Performance Computing Division.

Four years ago, Grider was working to decommission some old supercomputer hardware when it occurred to him there might be a better solution. "I realized our retired machines still had value since they all use Intel architecture these days," he says. "I had this idea that there ought to be a way to reuse these things. One way would be to help systems researchers."

The plight of systems researchers first appeared on Grider's radar screen at a supercomputing workshop, where a panel was asked how the government could help academics do better work on large-scale systems. The answer to that problem and the answer to Grider's disposal problem turned out to be the same: Recycle and reuse.

A new, one-of-a-kind computer systems research center called the Parallel Reconfigurable Observational Environment—or PRObE—has now been established to give systems scientists in academia unprecedented access to large-scale supercomputers.

PRObE is a joint effort of the LANL, Carnegie Mellon and the University of Utah along with the New Mexico Consortium, a partnership between the University of New Mexico, New Mexico Institute of Mining and Technology and New Mexico State University. Made possible through a $10 million National Science Foundation award, PRObE will eventually include at least two 2,048-core clusters to be housed in a research park near LANL, as well as smaller-scale clusters for early testing, including one located at Carnegie Mellon. All of these will be recycled machines donated by LANL.

The first large cluster is expected to come online in late spring.

Access to these clusters fills a pressing need felt by both systems researchers and computer science students, says Garth Gibson, a professor of computer science and electrical and computer engineering at Carnegie Mellon. High performance computing crossed the petascale threshold in 2008 with LANL's RoadRunner. Plans are already under way for the U.S. government to develop an exascale system by 2018 with 1,000 times more processing power than today's most powerful supercomputer. Google is rumored to already have a node count approaching a million spread across many data centers.

Unless they leave universities for government or industry jobs, Gibson says, researchers and students rarely have access to these expensive large-scale clusters. That means they don't get the training and education necessary to develop innovations for the fast-approaching era of exascale computing.

Moreover, when a supercomputer is new, it's immediately needed for applications research, he says, so even when they do get permission to use larger clusters, systems scientists can't run experiments on low-level hardware and purposely break these machines to see what happens.

For example, in massively parallel supercomputers with thousands of nodes, failure is a way of life, not an aberration. The key is developing systems that can continue performing well in a state of near-constant failure, Gibson says.

Researchers will be given dedicated use of the PRObE clusters for days, even weeks at a time. They will be allowed to replace any and all of the code and even inject faults that might be destructive to some equipment.

LANL isn't in the business of supporting academic research and didn't have authority to foot the bill to house, power, air-condition and maintain these old systems, Grider says. "We run a supercomputer complex to do nuclear weapons calculations," he says. "This is an offshoot thing that isn't in our mission." >>>

From page 3

That's why the New Mexico Consortium—an independent nonprofit managed by three state research universities—was called upon to help put together an application for NSF funding.

The NSF saw value in PRObE immediately, Gibson says, but it took some time to garner support for this large, unsolicited proposal. In that process, Carnegie Mellon was asked to lend its expertise to the project as a renowned leader in computer systems research. And the Flux Research Group at the University of Utah joined the team to adapt its powerful Emulab software to manage the PRObE testbed.

"Emulab is already used to manage about 40 network testbeds, but PRObE will be a unique facility," Utah computer scientist Robert Ricci says. "We're excited to be a part of this effort because it adds important new resources to the public research infrastructure."

If the PRObE pilot is successful, Grider says, it will provide high visibility validation of the need for large-scale systems research in academia and could serve as an example to be replicated by other government agencies.

PRObE also will conduct a summer school to train university students in how to build and manage very large high-performance computing environments; top students will be invited back to the center and LANL as interns.

"I would like to see a whole new generation of computer scientists that have some experience with computer systems research at scale," Gibson says. "Right now they don't begin to get the necessary training to understand the hard problems."

Jennifer Bails is a Pittsburgh-based freelance writer who frequently contributes to The Link, Carnegie Mellon Today and other publications.

It's All in the Game

With Foldit and EteRNA, computers and humans work together to crack genetic codes—and the results are being translated into real laboratory experiments

> **By Ken Chiacchia**

Human brains are becoming part of a vast, extended computing network that's creating new molecules of ribonucleic acid—RNA, one of the building blocks of all known forms of life.

They're doing it through EteRNA, an online program that pools players' ingenuity and then translates their insights directly into laboratory experiments. Launched in January, the game was designed by Adrien Treuille, an assistant professor in Carnegie Mellon's Robotics Institute, along with physicist Rhiju Das of Stanford University and Jeehyung Lee, a Carnegie Mellon computer science graduate student.

Cells in all living creatures are predominantly comprised of proteins that must fold into three-dimensional shapes in order to carry out vital functions. Understanding how proteins fold is central to understanding how they work and how they can be used to create favorable interactions within cells.

The recipe for each protein used by cells is encoded in DNA—deoxyribonucleic acid. Biologists long thought that RNA was a simple messenger that translated that code into the proteins that express genes, but recent research has shown that RNA can also have important catalytic functions, filling the normal role of proteins; and it can have regulatory functions, interacting with the genes in a distinctly DNA-like manner.

One of the reasons why RNA is so promising as a bioengineering agent is that it can affect cellular processes in multiple ways. Proteins that fold incorrectly can lead to diseases, but even "good" folding can sometimes be harmful. For instance, we don't want HIV proteins in an infected cell to fold correctly. That's why geneticists may want to block the action of certain proteins.

In EteRNA, a player begins with a target molecular shape and then tries to deduce the sequence RNA subunits (called bases) that would cause a protein to naturally fold into that shape. The goal is the creation of new molecules that might chemically block a virus from binding to its host cells, short-circuit a pathway necessary for a genetic disease to develop, or catalyze a new or improved industrial process.

Arguably, EteRNA's most significant innovation is that the gamers' work feeds directly into wet-lab research. On a weekly basis, the online community picks the most promising structures, which biochemists then synthesize and test.

Marvels Treuille: "And they're really just doing it because they want to beat their neighbor at some game."

Two earlier collaborative efforts were important predecessors of EteRNA. One was the SETI@home screensaver, which searched for extraterrestrial radio messages using volunteers' surplus computer time. Another was a program called Rosetta, designed by a team at the University of Washington led by biochemist David Baker.

Like SETI@home, Rosetta harnessed the surplus computer power of many volunteers, but instead of searching for radio messages, it calculated theoretical protein structures from their amino acid sequences—the "inverse problem" to what EteRNA does. Rosetta used the distributed computing power to calculate theoretical protein structure solutions quickly and displayed its solutions on participants' screens. But they weren't necessarily the best solutions.

Then something interesting happened. The participants, who had passively been watching the program fold amino-acid chains, told the researchers they thought they could improve on the structures they were seeing. That left the researchers wondering how they could harness the minds of these thousands of users to help find better solutions.

There's a precedent, of course, and it's right at Carnegie Mellon. Luis von Ahn, assistant professor of computer science at CMU, co-developed the familiar CAPTCHAs that use human image-recognition capacity to authenticate web users. >>>

From page 4

After calculating that people were spending about 500,000 hours per day interacting with CAPTCHAs, von Ahn decided to see if they could make more productive use of that time. He invented reCAPTCHA, which authenticates people by making them type two words—one a security test, the other a digitized word that computers have failed to identify. In the process, users are now helping to digitize thousands of books and newspaper articles that would otherwise be unsearchable in online databases.

Similarly, von Ahn took advantage of human image-recognition capability and created the ESP Game, which matches pairs of online gamers to produce searchable labels for images.

Taking a page from von Ahn's work, Baker teamed up with Treuille and Zoran Popovic, associate professor of computer science and engineering at Washington and a Carnegie Mellon Ph.D alumnus, to design a game called Foldit. The new game applied human problem-solving capability to the protein-folding task.

In an August paper in Nature, the researchers compared Foldit players' blinded solutions to Rosetta's for select known protein structures. The humans came closer to the real structures than the computers. More important, perhaps, was that human players employed distinct and more diverse strategies than the computers. Unlike the machines, players' strategies also changed between the early, mid-, and end games. This poses lessons for artificial intelligence design, Treuille says.

EteRNA differs from Foldit in three important ways: it goes from a target structure to a sequence rather than the reverse; it uses RNA rather than protein structures; and it creates a much more direct link between the computer simulations and real experiments done in biochemistry labs.

"Generally speaking, inverse folding is more interesting, because it allows you to design new things rather than just predict shapes," Treuille says. "But you have to understand how folding works before you can solve the inverse folding."

RNA's relative simplicity also should make EteRNA easier to learn than Foldit. And because it allows the inverse-folding approach, RNA should prove far more flexible in creating tailor-made structures with useful functions.

The competition between players in both EteRNA and Foldit can be intense. Interestingly, so can the collaboration. Foldit players have produced and shared widgets that carry out minor folding tasks, tips on strategizing the game and encouragement via a wiki.

Shawn Douglas, a Wyss technology development fellow at Harvard University and an expert on using DNA as a molecular building material, believes such games can help democratize science.

While professionals will still play a central role, amateur involvement "kind of raises all the boats," Douglas says. "People feel better about funding science, and there are more people who are educated about what's going on in science."

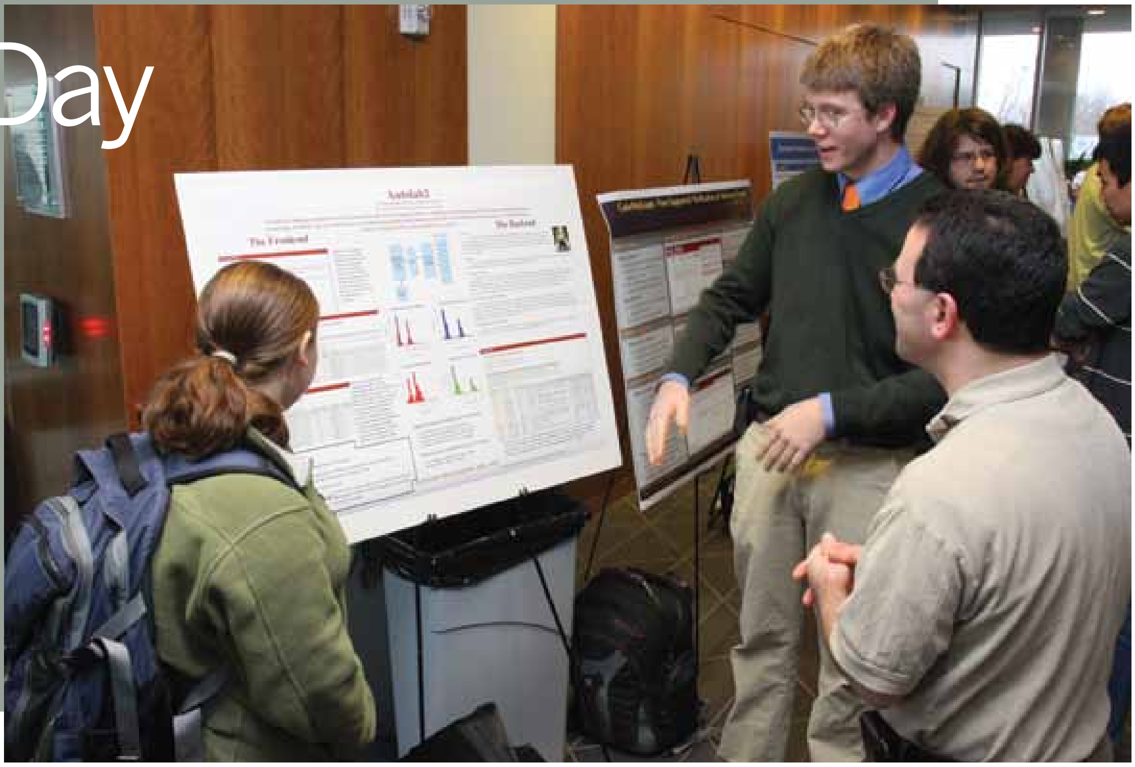
"I think the biggest question is ... what can we do with this?" von Ahn says. "Humanity's really large scale achievements—the pyramids, the Panama canal, the moon shots—all involved about 100,000 people. Before the Internet, coordinating more than [that] was just about impossible."

But reCAPTCHA is now using the brains of 750 million people daily—a little over 10 percent of the human race. Now that Foldit and EteRNA show that you can use online gaming to crack problems far more complex than digitizing text, the potential seems vast.

Editor's Note: You can access the EteRNA game at http://eterna.cmu.edu. Ken Chiacchia is an award-winning Pittsburgh-based writer who frequently covers medical and scientific topics.

Poster Day

Hunter Pitelka, senior computer science major, discusses his undergraduate research project with SCS senior Elizabeth Kemp and Greg Kesden, associate teaching professor, during CS Education Day Dec. 8 in the Hillman Center for Future-Generation Technologies on the Pittsburgh campus. The keynote speaker was Jan Cuny, program officer for the CISE Broadening Participation in Computing Program at the National Science Foundation.



WADE H. MASSIE PHOTO

What Does a Computer Believe?

If it's NELL, it knows what it "reads" on the web ... and then it tweets about it.

By Tom Imerito

Can a computer system form beliefs? Carnegie Mellon's Never Ending Language Learner does. More than half a million beliefs, in fact—and still growing.

Created by a research team led by Tom Mitchell, head of the Machine Learning Department, NELL autonomously and continuously reads the web; compiles words and their relationships to each other into a knowledge base from which it formulates beliefs; and then tweets its thoughts to more than 1,700 Twitter followers. It uses the words, "I think" when it tweets a new belief, whether mundane ("I think "ground cayenne pepper" is a #Condiment") or profound ("I think "art wedding photography" is form of #VisualArt").

The project is funded by the Defense Advanced Research Projects Agency, the National Science Foundation, Google, Yahoo! and Brazil's National Council for Scientific and Technological Development. The research goals behind NELL are creating machine learning programs that can run autonomously for decades; improving machine understanding of human language; and building the world's biggest digital knowledge base.

When NELL was launched in January 2010, its ontology (a collection of words and associations) included 123 noun categories and 55 possible relations between them. It's since grown to 500 categories and relations that include more than 20 million noun phrases and 50 million metadata phrases.

NELL's recent success comes after three years of effort, stymied in part by what Mitchell calls the tendency of relatively small ontologies to produce inaccurate results. Ambiguous and irrelevant words were periodically sucked into NELL's original, or seed, ontology. That produced increasingly inappropriate associations that multiplied in scope every time NELL read its 500 million web pages (a process that, amazingly, only takes about four hours). Mitchell calls them "avalanches of inaccuracies."

Tom Mitchell

A breakthrough came in late 2009 when Mitchell and his collaborators, including Burr Settles, a post-doctoral fellow in the MLD, and William Cohen, an associate research professor, had the idea that NELL might perform more accurately if they gave it more to do rather than less. They were right. They gave NELL a larger ontology which essentially gave the polluting or irrelevant words their own categories—and the avalanches stopped.

Settles illustrated the phenomenon with an example. "At first NELL confused spoken languages with programming languages, so it thought Fortran was a human language," he says. "When we gave it a programming language category, pollution of the language category stopped. The more (categories) we add, the better job it does."

NELL reads the web using a method called macro-reading, which analyzes associations by looking at patterns of phrases; the structure of sentences; the occasions when certain words appear together; and the surface structure (or "morphology") of specific nouns. Every day, NELL reads and rereads a local collection of 500 million web pages that are periodically crunched by Yahoo!'s 4,000-node M45 supercomputer to enable local analysis and processing. (The local collection represents about 10 percent of all pages on the web.)

To improve its accuracy, four individual learning components—each working on a different principle—analyze NELL's web reading. The multiple methods minimize the likelihood that any two components will validate an erroneous belief. One module scans NELL's resident 500 million local web pages for word and phrase patterns while another issues queries to Google, based on that initial reading, and extracts information from the web in real time. A third module looks for new rules based on the patterns of existing ones. The fourth module analyzes word morphologies. For example, the suffix "-ing" on a noun often indicates the word being described is an activity or hobby. If a noun is preceded by the word "Mount," the resulting phrase is likely to describe a mountain peak.

Data from each of the subsystem modules are then compared for consistencies and sent to NELL's Knowledge Integrator as "Candidate Facts," which are run back through the subsystem components for validation. If they survive all of the validation processes, they become one of NELL's "Beliefs."

Once accepted, the "Beliefs" are assigned confidence levels ranging between 50 and 100 percent. Those below 50 percent are discarded. To improve its confidence levels, NELL can re-read its web pages, query Google or ask its Twitter followers for verification. Soon, NELL also will use its "Beliefs" as the basis of an online game called "Polarity." The multiple choice game, developed by Settles, computer science professor Luis von Ahn and graduate student Edith Law, asks two players to categorize a low-confidence word served up by NELL. The game compares player responses to NELL's own assessment of the word and decides where and how well it fits into the knowledge base.

At the heart of NELL's ability to read and learn is bootstrapping—the process of discovering new categories, relations and rules in response to recurring word, phrase and sentence patterns.

Mitchell gave one example: "NELL read that the Mets played against the Braves and that the Mets play baseball. Therefore, it believes that the Braves play baseball. In itself that's not surprising, but the fact that NELL discovered it on its own is amazing. I think the point will come where NELL will be discovering things that we weren't aware of."

Mickey Moves In

The university's research bonds with The Walt Disney Company are getting stronger with a shift to the CIC.

> **By Tom Imerito**

In 1928, a struggling animator from Kansas unveiled the first animated cartoon to feature sound that was synchronized to the action on the screen. Walt Disney's "Steamboat Willie" was a blockbuster, and his name has defined the cutting edge of entertainment technology ever since.

More than eight decades later, scientists at Disney Research's two-year-old Pittsburgh laboratory—a collaboration between The Walt Disney Company and Carnegie Mellon—are using novel sensing technologies, human-computer interaction, robotics, computer vision and speech recognition to invent the world's next big entertainment experiences.

Researchers at the lab are investigating problems ranging from how to make a hand-drawn, animated dancer's dress whirl as though she were actually pirouetting in a dress made of real cloth; to designing algorithms that allow robots to learn activities such as tai chi. They've even got a video touch screen that can touch you back.

Disney Research has six laboratories around the world, but only two—the Pittsburgh and Zurich locations—are partnerships with universities. The late Randy Pausch (CS'88), a professor of computer science and human-interaction design at Carnegie Mellon, was instrumental in laying the groundwork for the Pittsburgh lab, says Jessica Hodgins (CS'89), professor of robotics and computer science and director of Disney Research Pittsburgh.

"Carnegie Mellon's collaborations with Disney originated with Randy Pausch's sabbatical at Disney Imagineering, where he worked and established ties before coming to CMU in 1997," Hodgins says. "Disney established two fellowships in Randy's memory for graduate students who bridge the arts and technology as he did in his research and teaching."

Currently located in the former Graphic Arts Technical Foundation Building near the Pittsburgh campus, the Disney lab is slated to move in early 2011 to the Collaborative Innovation Center. Disney is taking over a 17,000-square-foot space recently vacated by Google's Pittsburgh lab; the websearch giant has moved to a larger location not far from Carnegie Mellon in the city's East Liberty neighborhood. At the CIC, Disney researchers will bump elbows with scientists from Apple and CMU's Software Engineering Institute, among other tenants.

"We're very excited about the move to CIC," Hodgins says. "The additional laboratory space for our research in combination with the on-campus location is ideal for our many collaborations with faculty and students." >>>

DISNEY RESEARCH PITTSBURGH PHOTOS

Invented and developed at Disney Research Pittsburgh, TeslaTouch is a touchscreen that can provide tactile feedback to its users. One of a network of research facilities operated by the Walt Disney Company, the Pittsburgh lab is moving into Carnegie Mellon's Collaborative Innovation Center.

For the university, the Disney collaboration gives students and faculty the chance to work with researchers from the world's largest entertainment and technology conglomerate.

From page 7

Disney officials say the goal of the company's Pittsburgh lab is to develop advanced research in artificial intelligence, machine learning, humanoid robotics, speech recognition and human-computer interaction that can be used to develop products and content in a variety of Disney business units. For the university, the Disney collaboration gives students and faculty the chance to apply their theoretical knowledge to real-world problems and data, and to network with the world's largest entertainment and technology conglomerate. Two years into the partnership with Disney, SCS students and faculty are publishing cutting-edge research in leading journals around the world, in collaboration with researchers at Disney.

In an effort to make video games more true to life, a team led by Hodgins and Adrien Treuille, assistant professor of CS, along with Disney researchers Edilson de Aguiar and Leon Sigal have developed a new algorithm for modeling the dynamics of clothing so that it can be computed in real time. Other papers from the laboratory have looked at the importance of correct synchronization between audio and video and how to adapt motion capture data to characters such as a dancing penguin that are far from the human form.

But Disney Research Pittsburgh looks beyond the visual. Researchers have designed a touch screen interface that vibrates in response to a user's finger touch, providing instant tactile feedback. "The system provides an electrovibration stimulus to the finger via a transparent electrode placed on top of a glass screen protected by an insulating layer," Hodgins says. The development team includes interns and postdoctoral researcher Ali Israr under the direction of senior research scientist Ivan Poupyrev.

In robotics, a senior research scientist, Katsu Yamane, has developed algorithms that use motion capture data to program free-standing humanoid robots. "This optimization approach adapts the motion of a human actor to match the dynamics and joint limits of the robot," Hodgins says. "Techniques such as these allow robots to be programmed more rapidly and with a much broader range of behaviors."

Hodgins says that remodeling the CIC space will provide additional meeting rooms, laboratory space and offices for researchers and interns. More importantly, moving closer to SCS' other buildings will improve opportunities for collaboration.

Just as other research partnerships between private companies and Carnegie Mellon have been made stronger because of their proximity, moving onto campus is "essential" to the future of the Disney lab, Hodgins says.

Tom Imerito is the founder of Pittsburgh-based Science Communications. His work appears in magazines such as Automotive Engineering, PA Manufacturer, Pittsburgh Quarterly and Research Penn State. He also writes a column for the Pittsburgh-based technology and business journal TEQ.

Bob Murphy

Robert F. Murphy is the founding director of Carnegie Mellon's Lane Center for Computational Biology and the university's Ray and Stephanie Lane Professor of Computational Biology. A graduate of Columbia University and the California Institute of Technology, he joined CMU in 1983 as an assistant professor of biological sciences.

Beginning in the mid 1990s, Murphy and his team pioneered using the methods of machine learning to analyze microscope images of cellular structures.

At CMU, he developed the world's first formal undergraduate program in computational biology in 1987 and served as founding director (with Jelena Kovacevic) of CMU's Center for Bioimage Informatics, as well as founding director (with Ivet Bahar) of the joint Ph.D. program in computational biology offered by CMU and the University of Pittsburgh.

Murphy's honors include his election in 2007 as a senior member of the Institute of Electrical and Electronic Engineers and the Alexander von Humboldt Foundation Research Award in 2009.

He's also an avid amateur basketball player and a former youth basketball coach. He spoke to Link Managing Editor Jason Togyer.

How did you come to the field of computational biology?

When I was 13, I read a book called "The Genetic Code" by Isaac Asimov, and from that time, I knew I wanted to do something related to research in genetics. But I didn't know how I wanted to get there.

I ended up majoring in biochemistry. When I went to Caltech for my doctoral degree, I was looking for a way to analyze results, and I learned about the world of computing. I was hooked.

I spent a good fraction of my graduate school time doing data analysis, which in those days included actually figuring out how to get the data into the computer. About 50 percent of my time was spent doing experiments, and 50 percent was spent writing code. Since then, I have followed and learned the amazing developments in computer science and machine learning as they developed over the past 36 years.

What shaped your research interests?

I started having these odd experiences at conferences. My own work was very quantitative, but I would often present in a session with people who would be describing their results in very qualitative terms—and the images they were displaying, which were generated through microscopy, were supposed to be accepted in support of the models they were describing.

But I had a hard time making the connection between the images and the hypotheses they were attempting to prove. There was nothing that I would see that allowed me to use the word "proof."

Then I would see very similar images used to "prove" very different hypotheses. There was no attempt to deceive people, but the value of these images was woefully inadequate.

I said to myself, "Somebody has to tackle the method of drawing statistically verifiable conclusions from these images," and at a certain point, it became apparent that that person was going to have to be me.

I began working to develop computer models that would recognize patterns in the images, which in turn would enable researchers to say, "This particular protein is in this particular location."

How do you describe computational biology to a layperson?

Most of the time, I describe it as using computers to solve biological problems, and I say "computers" rather than "computer science" because people understand the concept of "computers." I also say I'm trying to change the way that biology is done.

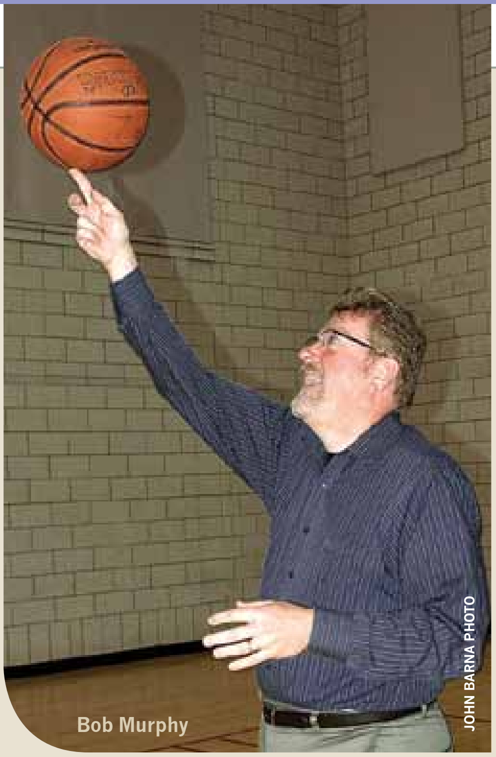
Using computers to solve biological problems is consistent with a traditional scientific discovery model—you have some data source, you analyze it and you report your results.

But the way that biological research and ultimately clinical practice will have to be changed is by having machine learning techniques take a role not just in analyzing data, but in collecting it as well.

The mission of the Lane Center is to enable—to catalyze— a transition to where robotics and machine learning are at the center of how biological research is done.

You use the term "active learning" to describe some of the work being done at the Lane Center. What does that mean?

Active learning describes an iterative process where a computer analyzes the elements of a



Bob Murphy

JOHN BARNA PHOTO

dataset, tries to build a model to predict the results of experiments that haven't been done and chooses key experiments to generate new data with the goal of improving the model until it can accurately predict all results without doing all possible experiments. In some sense, it removes traditional hypothesizing from the mix.

This is a very informative point to make—in traditional methods, you want to pick a hypothesis and prove that hypothesis is right. In active learning, you don't want to verify the hypothesis that you're already pretty sure is right—you want to test the hypotheses that aren't right, because those are the ones that will help you improve the model. Verifying hypotheses in which you already have high confidence isn't going to help you improve the model.

Why are these models more useful than, say, performing clinical trials on real patients?

Well, you might run clinical trials to see, for example, whether you have a statistically significant difference between "drug" and "no drug." But sometimes the effect is very small—maybe a 3 percent change. And the issue of side effects isn't examined until after a drug is approved.

Or, sometimes you have studies on one drug in one biological pathway that are run in parallel to studies of other drugs in other biological pathways. But the two studies don't inform each other, and those two drugs in combination create side effects. There are too many possible experiments to do.

That's where we see the need for ways to design the appropriate experiments to collect enough data to support much more thorough avenues to questions of whether a particular drug is an appropriate treatment for a particular disease. One of our goals now is to create detailed models of tissues that can predict effects, without necessarily measuring them.

Why is this work important to a doctor—or to a patient?

I'll give you the standard answer a basic scientist will give you—we're trying to learn the ways that clinical practice can be improved.

The FDA right now will not approve a drug without receiving a clear understanding of how the mechanism works. That concept comes from a time when we thought biology was much simpler than it actually is. Biological systems are incredibly complex, and therefore being able to make statements about "why something works" from a mechanistic standpoint is extremely difficult to do.

With machine learning techniques, we can create and evaluate models of drug efficiency from a sound statistical basis, without having to reduce it to a simple statement of "this drug affects the catalysis of A to B," because the model may determine the drug actually has 13 different effects.

What motivates today's students to pursue computational biology?

A lot of students are motivated because of things they see or read in the news. We're in an era when the opportunities for biomedical research—while also tackling significant computer science challenges—are enormous. And let's face it—the way that we humans work is a fascinating subject for us.

So we've been looking at the educational offerings of the Lane Center, and we already have two new initiatives there. The first is a master's of science in biotechnology, innovation and computing, which we're offering jointly with the Language Technologies Institute.

And we've initiated a new minor for undergraduates in computational biology, which is currently working its way through the review process. We're planning to bring that online in fall 2011.

BRAD MORRIS PHOTO

Closing the Educational Gap

A new report co-authored at Carnegie Mellon reveals an alarming decline in computer science education in U.S. elementary, middle and high schools. Several programs developed at CMU are helping to reverse the trend.

> **By Meghan Holohan**

The boxy robot chugged across the playing field with its target in sight as its designers watched with pride.

The designers—a group of teenagers, including 15-year-old Brendan Meeder—had programmed the Lego robot to knock black-and-white Ping-Pong balls off of a ledge, and then collect them with its mechanical arm. They'd tested their programming and engineering thoroughly. Everything was going according to plan.

It wouldn't have been an easy task, even if the robot were alone on the field. Unfortunately, it was playing Botball against other robots that had also been programmed to collect Ping-Pong balls. The team whose robot collected the most balls would win. Now another robot rushed across the playing field, cutting off the robot built by Meeder's team. And then a third robot that resembled a heavy cart rammed into the shelves, knocking the balls out of reach of its competitors.

Suddenly the easy-to-solve problem was a lot more complex. But while Meeder's robot might not have snagged the most Ping-Pong balls, a more important thing happened during that contest 10 years ago—computer science snagged Meeder.

Meeder was participating in the annual "Andrew's Leap" program run by Carnegie Mellon's School of Computer Science for high school students from throughout the Pittsburgh area. Since 1990, these students have come to campus for six to seven weeks to take classes with SCS faculty, visit their labs and work on projects in computer science and robotics.

Before Andrew's Leap, Meeder says he felt "adrift." Though he excelled in (and enjoyed) his math and science classes, he was unsure what he would pursue in college. After spending nearly two months at Carnegie Mellon learning about sensors, mechanics, programming and computer science theory, Meeder knew what he wanted to be when he grew up—a computer scientist.

Steven Rudich, professor of computer science and a co-founder of Andrew's Leap, says the program is designed to provide both education and mentoring to awaken and encourage students' interest in technology. "We want to stimulate them, to give them a deeper mathematical interest, give them more on the empirical level, give them more on engineering and introduce them to the many different areas of computer science," Rudich says.

Meeder (CS'07), who's currently pursuing his doctorate in the Computer Science Department, calls Andrew's Leap "the most important event that shaped my interest" in CS. "Before that, I just broadly enjoyed math or science," he says. "But after this experience, I really discovered what computer science is—and it is not something that is taught in a high school computer science class."

Meeder's right. The skills and concepts taught to 30 or so teenagers every summer during Andrew's Leap are not those being taught in computer science classes in Pennsylvania high schools—or in the other 49 states, for that matter. According to a new report called "Running on Empty: The Failure to Teach K–12 Computer Science in the Digital Age," high school computer science classes tend to focus on the drudgery of programming languages, not on the fundamentals of computer science. Students get frustrated and decide that computer science is drudgery.

That's if they get any exposure to computer science at all—the report, co-authored by CMU's Mark Stehlik and Leigh Ann Sudol, along with Cameron Wilson of the Association for Computing Machinery and Chris Stephenson of ACM's Computer Science Teachers Association, notes that many schools, facing budget pressures and federal mandates created by the No Child Left Behind Act, have dropped computer science education altogether.

As a result, conclude the authors, computer science education in the United States is suffering right at a time when computer science jobs and >>>

BRAD MORRIS PHOTO

Opposite page and above: Students compete in the 12th annual FIRST Lego League tournament, held Dec. 4 at CMU's National Robotics Engineering Center in the Lawrenceville section of Pittsburgh. Run by the Robotics Academy, the tournament is one of several Carnegie Mellon initiatives designed to excite K-12 students about careers in math, science, engineering and computer technology.

"The biggest resistance I get is that people say 'you can't teach 13-year-olds computer science.' But you can. Kids can learn it. Even a student with a second-grade reading ability can look inside a computer and understand that information is traveling through the wires as electrical pulses."

Tammy Pirmann

ideas are driving the global economy. Written for the ACM and CSTA, "Running on Empty" was designed to get a look at the "state-by-state landscape," says Stehlik, SCS assistant dean for undergraduate education. The report concludes that landscape isn't pretty.

A variety of programs developed at the School of Computer Science are trying to improve the view. Some include direct outreach to students in kindergarten through 12th grade, such as Andrew's Leap, which just celebrated its 20th anniversary; and a new effort called FIRE, for "Fostering Innovation through Robotics Exploration." Other programs, such as CMU's Computer Science for High School, or "CS4HS," are training the next generation of K–12 teachers how to incorporate computer science into their existing math and science curricula.

According to "Running on Empty," those programs and others like them are needed more than ever. Stehlik and Sudol, a Ph.D. student in the Computer Science Department, report that public schools in only 14 states offer computer science courses at levels recommended by CSTA and ACM. Two-thirds of states don't require even one standard upper-level computer science course in their curriculum for high school students.

All states mandate a core curriculum consisting only of English, math, social studies and basic science, just as they have for generations. In 40 states, students aren't even allowed to count computer science classes toward the number of credits in math or science required to graduate.

As a result, Stehlik says, fewer students are graduating high school with any computer science background. He points to two causes for the decrease—the federal No Child Left Behind Act, and changes in the way that advanced placement courses in computer science are handled for high school students.

Launched by the School of Computer Science in 2006, Computer Science for High School is an annual three-day summer workshop for teachers that provides classroom resources and curriculum strategies for integrating computer science concepts and principles into their lesson plans. Topics include robotics, computational biology, web search strategies and computational thinking. The 2010 CS4HS workshop was held July 26–28. Sponsors include CMU's Women@SCS Program, Microsoft Research and Google.

Much ink has been spilled over issues created by No Child Left Behind, which was signed into law in 2001. The legislation requires all K–12 students to demonstrate basic knowledge and skills on standardized tests that focus heavily on math and English. Performance on those tests is tied to the amount of funding the federal government provides to states and individual school districts. Subsequently, critics say, schools are encouraged to "teach the test," gearing their curriculum to heavily cover a limited range of subjects in hopes of boosting their scores. Many districts have eliminated classes that aren't covered by standardized tests—such as computer science—and moved teachers from computer classes to basic math instruction.

Another problem, according to Stehlik, is that the College Board—the organization that sponsors advanced-placement tests for high school students—has changed the requirements for AP examinations in computer science. The College Board once offered two different computer science tests, including one that tested theoretical knowledge of algorithms, data structure and data abstraction—the fundamentals of computer science. In 2008, the College Board dropped that test, called "Computer Science AB," citing "declining interest" and a lack of funding to continue offering it. The remaining test only examines students' mastery of the Java programming language. Without an incentive to pass the more rigorous and interesting "AB" test, fewer students are motivated to take upper-level computer science courses.

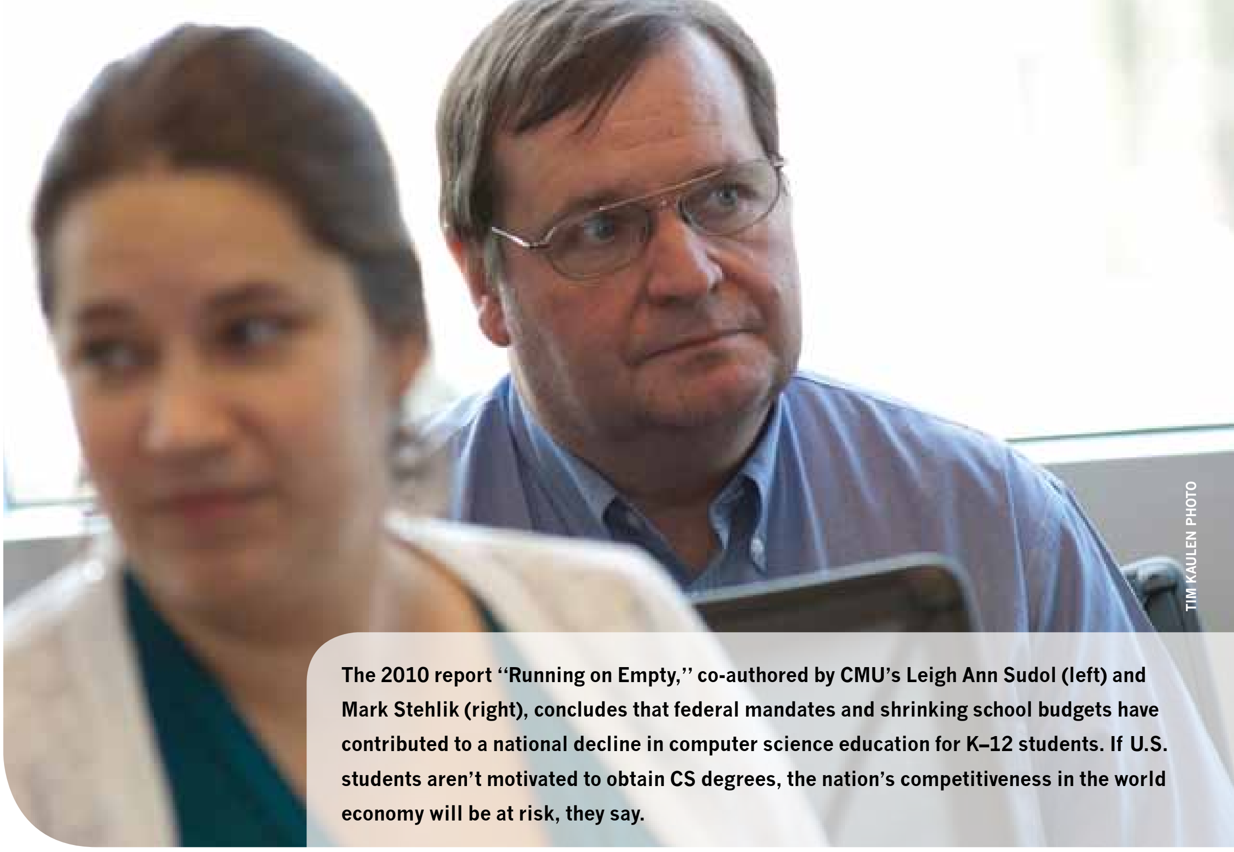
While Stehlik and Sudol paint an overall bleak picture, they do find some bright spots. In Texas, for instance, a group of motivated teachers convinced the state Board of Education to mandate tougher computer science education in its curriculum requirements. And individual teachers are also making a difference. In the Springfield Township School District near Philadelphia, teacher Tammy Pirmann and her colleagues petitioned the school board to make fluency in computer science, and not typing skills, a graduation requirement. As a result, the district has adopted the recommendations created by the CSTA for grades K–12. Now, by the time students reach Springfield Township High School, they've had eight years of computer science instruction.

"There are many kids who would have never taken a computer science class," Pirmann says. "They end up in a computer course because it's required, and they find out they're good at it—and they like it."

Pirmann teaches Computer Science in the Modern World, which every Springfield Township student must take in ninth grade in order to graduate. In Computer Science in the Modern World, students learn about hardware, networks, binary numbers, encryption and encoding, interface design and programming.

Of course, not every Springfield Township student is a computer science whiz by the time they get to ninth grade, or even when they graduate, Pirmann quickly adds. "I have the same cross-section of students that any school has," she says. But they can learn that computer science is a broader subject than programming, Pirmann says.

"The biggest resistance I get is that people say 'you can't teach 13-year-olds computer science,'" says Pirmann, who worked as a computer industry consultant before becoming a teacher. "But you can. Kids can learn it. Even a student with a second-grade reading ability can look inside a computer and understand that information is traveling through the wires as electrical pulses."



TIM KAULEN PHOTO

The 2010 report "Running on Empty," co-authored by CMU's Leigh Ann Sudol (left) and Mark Stehlik (right), concludes that federal mandates and shrinking school budgets have contributed to a national decline in computer science education for K–12 students. If U.S. students aren't motivated to obtain CS degrees, the nation's competitiveness in the world economy will be at risk, they say.

Springfield Township's programming unit uses the Alice graphical language developed at Carnegie Mellon, and uses the textbook "Learning to Program With Alice," authored by CMU associate professor of computer science Wanda Dann, Stephen Cooper of St. Joseph's University and the late Randy Pausch (CS'88), CMU professor of computer science and human-computer interaction.

Most school districts aren't like Springfield Township and have no computer science graduation requirements. Indeed, some have so many students struggling to learn the basics of reading and math that adding a rigorous computer science program isn't feasible. For those districts, there are materials such as those offered by CMU's Robotics Academy, which provides a variety of curriculum-boosting activities for schools. Robin Shoop, director of the Robotics Academy (part of the Robotics Institute), says schools can use as much or as little of the material as they want. "It's popular in education to use robots to teach STEM (science, technology, engineering and math)," Shoop says. But robots can be expensive, and what does a child do if she wants to continue to experiment with programming and robotics at home but doesn't have access to robots?

The answer is a virtual world featuring Lego-brand robot products. The software is being developed by the Robotics Academy; Ed Paradis, a senior research programmer at CMU's National Robotics Engineering Center in Lawrenceville, demonstrates. On Paradis' monitor, a Lego robot rolls across a beach as waves lap at the shore. Palm trees dot the landscape as the robot bounces across the terrain. Students collect coins—like in Nintendo's Super Mario Brothers from days of yore—while programming the Lego robot to navigate different environments, including the island and outer space. "The real benefit is that the students can do all the same things they do with (physical) robots, but they can do it in a lab without robots," Shoop says.

Paradis adds that the game is also designed to hold student interest: "It will be entertaining, like 'Where in the World is Carmen Sandiego'."

The game, which will be free of cost, is being developed under the auspices of a program called Fostering Innovation through Robotics Exploration, or FIRE. Funded with a four-year, $7 million grant from the Defense Department's Advanced Research Projects Agency, FIRE is a collaborative effort among several different units of the School of Computer Science, including the Robotics Academy, the Human-Computer Interaction Institute, the Robotics Institute and the Language Technologies Institute, as well as the University of Pittsburgh Learning Research and Development Center.

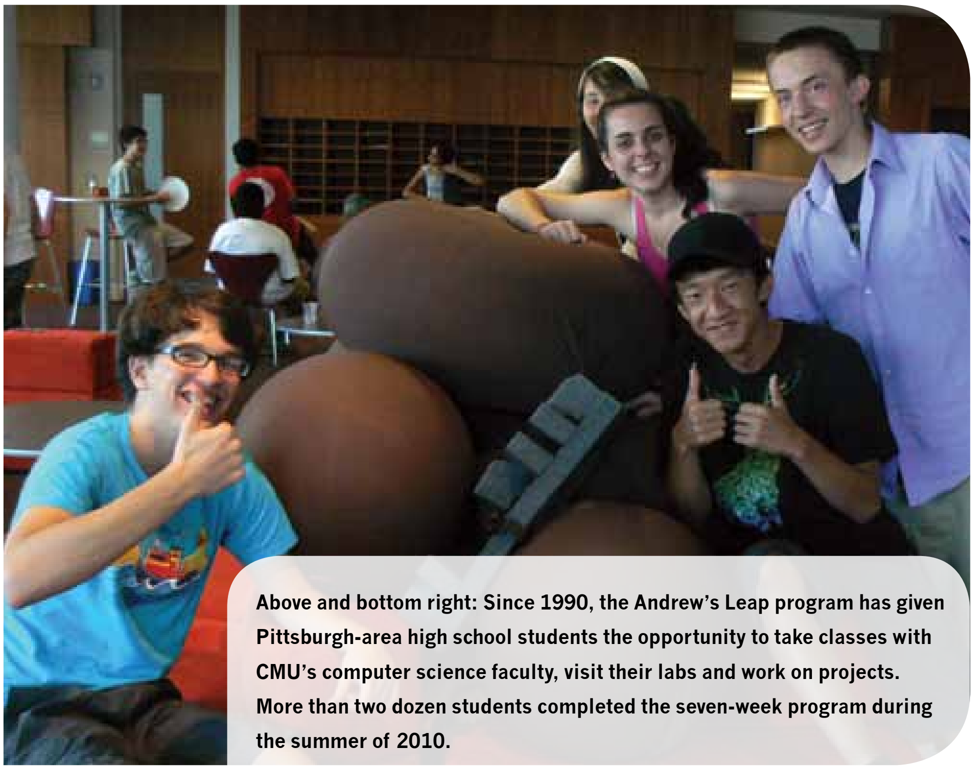
FIRE is developing tools like the virtual Lego Robot to enable middle- and high-school students to extend their interest in robots from one STEM activity to the next. Other tools being developed include computerized tutors that will teach math and CS skills in the context of robotics. The initiative targets robotics competitions that are already popular with secondary school students such as For Inspiration and Recognition of Science and Technology (FIRST), VEX and Robotfest.

FIRE also builds on the existing successes of the Robotics Academy, which since 2000 has developed educational materials and curriculum for computer science education in both middle and high school. Teachers can use the curriculum in ways that fit their needs.

Robotics Academy-designed software teaches computer science in different ways. In Robot Algebra, for instance, students answer basic math questions to control robots on a screen. The program is intended for students who have a hard time understanding ratios and proportions; if a student wants to get three robots to dance together, but the robots are different sizes and shapes, the student needs to choose the correct proportions so that the robots can interact without bumping into one another. The program is built around the Cognitive Tutoring technology developed at Carnegie Mellon, and provides hints and additional information to students to help them answer correctly.

Mike Dischner, a teacher at McKeesport Area High School just outside of Pittsburgh, uses materials from the Robotics Academy in his engineering class and the school's robotics club. Unlike Springfield Township, the McKeesport school district has not made computer science a graduation requirement. "If anyone wants to be exposed to computer programming related to automation or machines I'm the only ball game," Dischner says. "A lot of school districts are more worried about the students performing on standardized state tests, so (computer science) is just not a priority at this time."

Although teachers know that programming involves math, Dischner says that without >>>



CATHERINE COPETAS PHOTOS

Above and bottom right: Since 1990, the Andrew's Leap program has given Pittsburgh-area high school students the opportunity to take classes with CMU's computer science faculty, visit their labs and work on projects. More than two dozen students completed the seven-week program during the summer of 2010.

resources, "who's going to teach those types of things?" Materials from the Robotics Academy help close the gap. Dischner, for example, uses its curriculum in the engineering class he teaches for vocational technology students at the high school. Dischner also oversees McKeesport Area's FIRST Robotics Team 1708, which is open to all students. The team participates in regional robotics competitions and several alumni have gone on to study computer science in college, he says.

In addition to offering curriculum resources, SCS is also training teachers in computer science education. CS4HS—Computer Science for High School—is a three-day summer workshop for computer science teachers run in partnership with Google that provides ways to help teachers capture their students' attention. Lenore Blum, distinguished career professor of computer science at Carnegie Mellon, is one of the founders of the program. She says teachers can be "really effective" as change agents because they're in the classrooms with K–12 students all the time, and they know what works and what doesn't. "The spinoff or networking effect is enormous," Blum says. "The teachers who come here tend to be leaders and very professional, and they influence opinions in their communities."

Are the programs effective? Anecdotal evidence suggests they do make a difference. Pirmann, for instance, took up the challenge of implementing the CSTA curriculum in the Springfield Township district after she attended the CS4HS workshop, and she's since worked closely with Stehlik and other CMU faculty.

And efforts such as Andrew's Leap clearly have made an impact on students such as Meeder, who 10 years later can vividly recall playing a 3D version of a Star Wars game in Pausch's lab and thinking how cool it was to have access to such technology.

Still, those efforts are limited in scope, Stehlik says. "All of these things are important, but we're reaching small audiences," he says.

What's needed, Stehlik says, are changes to school requirements on the state level. When it comes to computer science education, Pennsylvania, for instance, ranks in the middle of the pack, according to "Running on Empty." In the Keystone State, teachers don't need a computer science degree to teach computer science; instead, they need a state certification to teach business classes. The requirements for computer science teachers were first created in 1981, and haven't changed since, Stehlik says.

For real change to occur within K–12 computer science education, motivated educators need to advocate for that change on the state level, just as Pirmann lobbied her school board. "We were working on a local level, trying to get computer science as a core subject, and we found we really need to talk about it at the state level," she says.

Pirmann, Stehlik and other educators recently approached the Pennsylvania Department of Education, asking for a change in graduation requirements that would include computer science credits, as well as changes in the way that computer science teachers are certified. They received a lukewarm reception. While the department acknowledged that computer science education was important, it was hesitant to change any regulations; Stehlik says "it's difficult to move 30-year-old machinery."

Federal, state and local officials have to stop "passing the buck" and work together to revamp computer science education in all grades, Stehlik says. With the decline of the nation's manufacturing base, computer science and high technology are the future of the U.S. economy, he says.

And if U.S. students aren't working on the next new blockbusting technology in computer science, students from other countries will. Stehlik points out that according to the U.S. Bureau of Labor Statistics, between now and 2018 nearly 75 percent of new American science and engineering jobs will be in computing fields. But only 50,000 American students are expected to obtain computer science degrees between now and then.

Stehlik wants students to be inspired to create the next "killer app" like Facebook or Twitter. "There should be a sense of wonder about what we can do with computer science," he says.

Meghan Holohan is a Pittsburgh-based freelance writer who frequently contributes to The Link, PittMed, MentalFloss.com and other publications. To download "Running on Empty," visit www.acm.org/runningonempty.

WADE H. MASSIE PHOTO

Stephanie Rosenthal (CS'07,'09), a Ph.D. student in computer science, demonstrates CoBot 1. The robot, under development in Manuela Veloso's CORAL lab, is designed to serve as a guide and appointment secretary and can navigate the CMU campus by calculating its proximity to wi-fi antennas.

Making a Mechanical Pal

Several new social robots are expected to start prowling the halls (and playing games) at CMU this year. But giving a robot personality takes a lot more work than just putting on a happy face.

> **By Jason Togyer**

A child who grew up watching TV in the 1960s, '70s and '80s would be forgiven for assuming that she'd have a robot pal by now.

After all, according to "Lost in Space," robot B-9 was always present to warn Will Robinson of danger in what was (in the 1960s) the far-off year of 1997. On "The Jetsons," Rosie the Maid was a helpful (if sarcastic) electromechanical household companion in the 21st century.

In his book "Your Flying Car Awaits," author and historian Paul Milo reports that it wasn't just TV scriptwriters who assumed that robots that work closely with people would be a feature of everyday life by the year 2011. Responsible, respected futurists working for organizations such as the Defense Department, the Rand Corporation and IBM also figured that by the 21st century, robotic butlers, cooks, chauffeurs and babysitters would be commonplace.

Indeed, writes Milo, some technologists were concerned that we'd have too many robots by now. They speculated that humans would either be thrown out of work or that they'd object to being served by robotic assistants. One researcher even suggested that chimpanzees be trained to take over jobs (such as driving cars!) that humans wouldn't want done by robots.

The problem of training monkey chauffeurs to take over from robots hasn't yet developed. And >>>

Robotics master's student Joydeep Biswas demonstrates the user interface of CoBot 2, one of the social robots under development in Manuela Veloso's CORAL lab. CoBot 2's telepresence capability will allow it to serve as a virtual stand-in for a human located anywhere in the world.

developing practical robots that work closely with people and which can respond using human communications methods—social robots—turned out to be a lot harder than futurists suspected two generations ago.

In fact, the creation of a social robot is still so new "it's almost a craft process," says Jodi Forlizzi, associate professor of human-computer interaction, who's part of CMU's Project on People and Robots.

But research into social robots has been ongoing since the 1940s, when American-British neuroscientist Grey Walter, a pioneer in the use of electroencephalographs to study brain waves, speculated that many of the functions of animal brains could be simulated by electrical components. By 1951, Walter had built crude but working autonomous robots that exhibited almost animal-like behavior; they reacted to noises and lights and could be "taught" rudimentary activities, such as "begging" for attention.

With these early successes in cybernetics—and with transistors and integrated circuits leading steadily to more and more powerful digital computers—it was natural to assume that robots such as Walter's also would get smarter and more humanlike.

"After Moore's Law was articulated, some scientists basically extrapolated present-day trends about the pace and increase in technology, and figured that within the next 20 years or so, they'd have robots that would be smart enough to take over these jobs," says Milo, speaking from his home in New Jersey, where he's currently working on a book about higher education.

"But it wasn't necessarily a case of making computers faster and smarter," he says. "There has to be some sort of 'quantum leap' that bumps us from one track to another. You can continue adding horses to a carriage, and you'll get a carriage that runs faster—but you won't have an automobile."

Indeed, processing speed isn't the limiting factor for determining whether a social robot can successfully interact with humans in their environment. Instead, the problems include those of detection, interpretation and communication—recognizing human beings and understanding what they need—and they have layer upon layer of complications. Not only do robots have to understand what humans are doing—the humans have to intuitively understand what the robots are doing, without the need for interpretation.

"Robots have all sorts of limitations in their social interactions," says Manuela Veloso, the university's Herbert A. Simon Professor of Computer Science. "Sometimes they may not understand what you say. Other times, they may not be able to complete a task. What social robots are able to do right now is very limited." Veloso and other roboticists are trying to expand the horizons for social robots and remove those limitations.

Veloso leads a research group called CORAL—for Cooperate, Observe, Reason, Act and Learn—that studies the way groups of autonomous robots can be programmed to work together on tasks, and teaches a project course in designing intelligent humanoid robots. She's also among the faculty members who will be conducting real-life tests of social robots on campus this year.

Veloso's CoBots (short for "Companion Robots") are designed to deliver mail and other items to campus offices and act as companions and tour guides to visitors. The CoBots also will have a wide area in which to experience human interaction. While some of CMU's early social robots were fixed in place or had limited mobility, the CoBots are speedy, and are designed to eventually roam around the Gates and Hillman centers and throughout the Carnegie Mellon campus. Other social robots that will be testing soon at Carnegie Mellon include Gamebot, which will be able to play Scrabble with individuals and groups, and Snackbot, which will deliver treats upon request.

Social robot research at CMU has a long history that builds on the legacy of pioneering research into artificial intelligence by Simon, Allen Newell and others. Today's projects are spiritual descendants of CMU's Social Robots Project, an interdisciplinary effort begun in the 1990s as a joint project of the School of Computer Science and the School of Drama.

The Social Robots Project set as its goal creation of robots that had "personalities" and which could be given tasks and interact with human beings according to social conventions. It eventually spawned "VIKIA" and "GRACE," which could engage in some of the same activities as a typical college student, including giving a presentation at a conference; and in 2004, "Valerie," CMU's first robot receptionist, or "roboceptionist."

The stakes for developing social robots are quite high. Lessons learned from projects such as Snackbot or Gamebot, for instance, could eventually inform work being done at Pittsburgh's Quality of Life Technology Center and lead to improvements in robots that perform rewarding tasks for society, including care for the elderly and handicapped, tutoring children and reaching out to people with developmental disorders such as autism.

Industrial robots have been a common fixture of the developed world since the 1970s, and robotic rovers have played an ever increasing role in fields such as space exploration and search-and-rescue operations. Commercially available robots such as the semi-autonomous Roomba vacuum cleaner have also become common.

Yet when the general public thinks "robot," they don't often imagine a disembodied arm that welds fenders or a scientific robot gathering samples on the surface of Mars. Instead, they usually picture a science-fiction robot such as C3PO of the "Star Wars" movies.

Speculative fiction about robotics has "layered on expectations," says Forlizzi, whose background includes work in industry as an interaction designer and as a researcher on new product development. She's currently part of several social robot projects underway at CMU, including Snackbot and the Home Exploring Robot Butler, or HERB, which is a joint investigation of Intel Research Pittsburgh and the Quality of Life Technology Center.

While both Snackbot and HERB are social robots, they couldn't be more different in appearance or purpose. Snackbot is child-sized (four-and-a-half-feet tall), enclosed in a smooth plastic housing, and has a round face with two "eyes" and a digital mouth. It has two arms, but they're fixed in place, designed to support a serving tray.

HERB is larger and more industrial in appearance, and has two highly mobile arms that can grab objects—such as canned goods or utensils—and bring them to a human. As a result, HERB is a more sophisticated robot, but it also has the potential to be off-putting, Forlizzi says.

"It's huge," she says. "Would you be comfortable with it in your home?" And because HERB is designed as a robotic assistant for people with limited mobility, such as those with spinal cord injuries, a task that sounds straightforward—like fetching an object—poses several serious challenges for its developers, including consideration of the social and emotional needs of the people who HERB will be assisting. It's hard enough for HERB to successfully navigate a kitchen or dining room, but it's also got to avoid sudden movements that seem alarming.

"If the robot just brings something to you and shoves it into your face, that's a little bit intimidating," Forlizzi says. "We have to find a better way to make it more social."

When interacting with humans, she says, robots have to move and communicate in ways that mimic polite human behavior. They need to meet a person's gaze, move in ways that aren't threatening, and avoid invading "personal space." If they communicate using spoken language, they need to understand when and how to interrupt someone.

"In order to have good social interaction, a social robot has to be aware of the context around it," says Reid Simmons, associate director for education at CMU's Robotics Institute and a research professor of robotics and computer science. "Let's say I'm a pill-dispensing robot, and a person is supposed to take a pill three times a day. If someone is napping, I probably shouldn't wake them up to give them a pill."

In other words, when a robot is placed in a setting with humans, it needs to act like a human, says Paul Rybski, systems scientist in CMU's Robotics Institute. "Usually, the more anthropomorphic you can make them, the easier it is for people to try to use their social communication skills to interact," he says.

Like Forlizzi, Rybski is a member of the team working on Snackbot, which is designed to incorporate as many off-the-shelf components as possible. It "listens" using a microphone that's designed for tele-conferencing applications and which can pinpoint the direction of the loudest voice in a room. It detects obstacles using laser sensors sold for industrial applications such as inspecting pipes or measuring distances.

The availability of those components enables robotics researchers to spend less time worrying about hardware and more time refining the software that predicts and interprets human behavior. But while the sensing technology has become less expensive—in part due to the widespread use of robots in industrial settings—interpreting the inputs is still difficult.

"All of these things that we take for granted in people, that we can see, that we can move around obstacles, that we can go from place to place—from a roboticist's point of view, just building a robot that can navigate an environment by itself is an accomplishment," Veloso says.

> **"Robots have all sorts of limitations in their social interactions. Sometimes they may not understand what you say. Other times, they may not be able to complete a task. What social robots are able to do right now is very limited."**
>
> *Manuela Veloso*

Adding a social interface compounds the difficulties. While a directional microphone can help a robot detect where the loudest noise in a room is coming from, it needs signal-processing software to determine whether it's receiving a spoken command or just hearing a passing conversation. Proximity sensors and cameras can "see" an obstacle blocking a robot's path, but it needs to be able to tell a person from a trash can. Simultaneously interpreting multiple inputs—detecting movement as well as noise, and determining whether >>>

the object moving is also the object making that noise—is another serious programming challenge, Rybski says.

Snackbot will have limited ability to engage in spontaneous activities. While it will be able to independently navigate corridors in Newell-Simon Hall, its more important role is to serve as a research platform for studying human-robot interaction over a long term.

The team is especially interested to see if people modify their own behavior around Snackbot after repeated encounters, Forlizzi says. Will they appreciate and understand Snackbot's user interface? Will they welcome the addition of Snackbot to their daily routine? To capture the information, Snackbot will make a video and audio record of its day-to-day activities that researchers can then mine for data.

"There's a lot we don't know yet about human interaction with non-human objects, and we don't have a lot of ways to get unbiased data on those interactions," Rybski says. "We need to study why people respond to one robot, but not another."

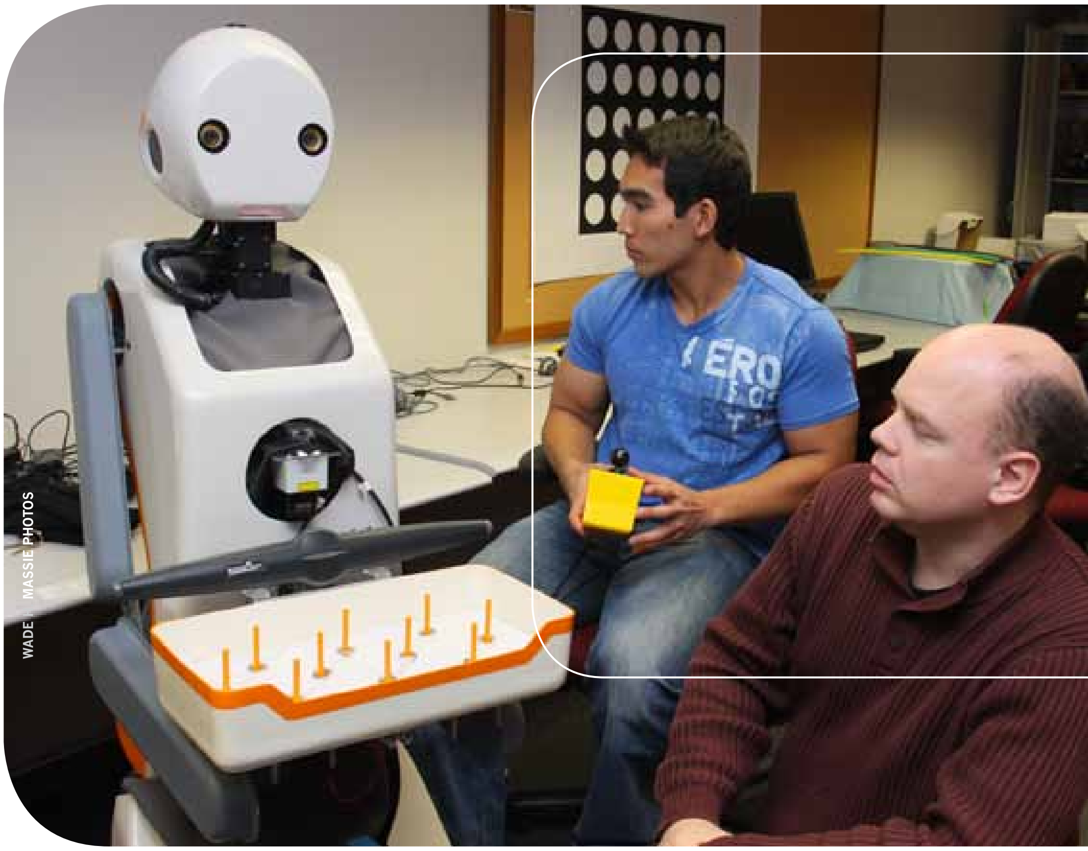
For social robots to be truly useful, interaction needs to be intuitive. "You have to be able to rely on someone's existing knowledge," Rybski says. The user of a robot such as HERB has to be able to talk to the robot using plain commands and then understand the robot's feedback immediately. "You can't hand them a 3,000-page manual or ask them to take a course," he says. "It's got to be able to interact with people in a way that they're comfortable with."

Unfortunately for Snackbot, Rybski says, "people are notoriously difficult to interpret and understand—just ask any human."

Ethnic background, native culture, gender, age, education level—all are factors in how people interact with one another, says Simmons, who calls humans "infinitely variable."

"When humans interact with each other, they can accommodate that variability," Simmons says. "A robot has a fairly limited range of things it can react to. The traditional view of interaction is turn-taking—I do something, and then you do something. But that's not really an accurate model of how people interact. It's more like a dance—we're constantly changing our interactions based on the feedback we receive."

Take the simple act of telling a joke, Simmons says. If the person telling a joke senses through non-verbal cues—an arched eyebrow, a disgusted or puzzled expression—that her listener is offended or doesn't understand, she can adjust the tale or stop altogether. "Gaze, gesture, posture are all incredibly important," Simmons says. And if something unexpected interrupts a conversation—a fire alarm, a scream, one of the participants suddenly fainting—humans would understand what to do, but a robot designed to "take turns" could be stymied.



WADE MASSIE PHOTOS

Simmons uses an example of the automatic checkouts now common in supermarkets and discount stores—they prompt the users to perform specific tasks, such as moving their purchases, even if the person's already done that.

Having a robot that can offer directions and guidance "in a way that doesn't annoy people is very important," Simmons says. Robots also need to be able to signal when they don't understand a task—either by saying "I don't understand," or by some non-verbal cue, such as a tilt of their "head."

Simmons was the lead developer of the Carnegie Mellon "roboceptionists" who have greeted visitors to Newell-Simon Hall since 2004. Though the "roboception" desk is currently occupied by a robot named "Tank," the first roboceptionist was the much-chattier "Valerie." Both were developed in cooperation with CMU's College of Fine Arts.

Experiences with Valerie and Tank gave researchers valuable insight both into human-robot interaction and into the creation of experiments involving social robots, Simmons says. "We had certain ideas about what social interaction with a robot would be like," he says, but their experiments quickly hit the limitations of a "receptionist" framework.

For instance, the developers expected that visitors would spend one-on-one time with the roboceptionist. Instead, they tended to approach the roboceptionist in groups of two or three. "But the robot doesn't understand group interaction," Simmons says. "If you ask it to tell you its name, it will tell you, but if another person in the group asks, it will say, 'I already told you my name.'" Valerie and Tank have no way of knowing that a different visitor was "talking."

The roboceptionists also lacked personalization. Though someone might pass the roboceptionist every day, the robots have no way of recognizing her, and treat her as if she's visiting CMU for the very first time. "That's almost the exact opposite of human-human social interaction," Simmons laments.

And tasks performed by the roboceptionist lacked scope—a visitor asks a question, and Tank provides an answer. Then the visitor moves on. That doesn't provide much time for roboticists to study the interaction process.

Simmons' new Gamebot project will incorporate several features not available in the roboceptionist. Gamebot will recognize and remember players' faces and will engage them in a significantly more difficult task—playing the word game Scrabble—that enables the roboticists to better study group dynamics.

"We chose Scrabble because it encourages multiple people to play, and it's a fairly long game, but you can leave Scrabble at any time, so you can play as much or as little as you want," he says.

Far left: Grad student Pong Sarun Savetsila and Paul Rybski, senior scientist in the Robotics Institute, run some tests on Snackbot. The robot, which will begin limited testing this spring, is designed as a research platform for studying human-robot interaction.

Left: A sign in Rybski's lab seems to imply that even robots need an occasional nap.

Gamebot will keep statistics on when and how often people play, adjust its own gameplay to accommodate a specific user and will recognize patterns—including if a player doesn't visit for a length of time. "The idea is that if we can personalize the interaction, will that make a difference in how people interact with the robot?" Simmons says. "Do people who get personal treatment tend to come back more than people who don't?"

The success of a social robot such as Gamebot can be evaluated in part by its ability to build, maintain and expand its relationship with humans over a period of time. But few social robots will need to lean on those relationship skills as much as the CoBots currently under development in Veloso's lab. Veloso, president-elect of the Association for the Advancement of Artificial Intelligence, says CoBots are designed to create a "symbiotic relationship" between humans and robots—while the robots will help humans, humans also will have to help the robots, which will be unable to complete certain tasks without assistance.

"I think one of the key ingredients for a social robot to be successful is to demystify it," she says. "As CoBot moves in the world, if it cannot perform a task, it asks for help."

Both CoBot 1, which was activated in 2009, and CoBot 2, which came online last year, are going to need winning personalities, because there's nothing overtly cute or vaguely anthropomorphic about either one. ("To me, it's a robot," Veloso says. "I'm not trying to make it nice and pretty—I'm trying to make it functional.") Built on wheeled omnidirectional platforms designed by Mike Licitra of CMU's National Robotics Engineering Center, the two CoBots don't have faces or arms. They can receive commands verbally or from a keyboard.

"CoBot can't lift objects, and it can't press buttons," Veloso says. That's a challenge for a robot that's supposed to deliver mail and other items, and escort visitors from place to place, but it's a limitation that Veloso has embraced. CoBot provides opportunities to learn how social robots should approach humans for assistance, and also to determine how social robots should fail at tasks—for instance, how to behave when they can't understand a request.

CoBot 1 navigates by calculating its proximity to the wireless network antennas that are common sights on campus; CoBot 2 finds its way using Hagisonic's StarGazer robotic navigation system, which requires placement of a series of adhesive dots along hallways and other passages.

"We've tried to enumerate different tasks that could capture different problems that need to be solved—scheduling, navigation, identifying visitors," says Veloso, who is planning to create a total of 10 CoBots over the next five years. Users will be able to request certain tasks—escorting a visitor from place to place, fetching a parcel—through a web interface. Ultimately, the CoBots also will be able to communicate with one another to divide tasks.

Veloso says it "doesn't make sense" to develop just one CoBot. "We're not in the business of interacting with one person at a time, but with many people," she says. "So we should have many CoBots."

Future areas for exploration include developing the ability of CoBot to adapt to unexpected input, Veloso says. "Right now, if it's moving down the corridor and you say, 'Hello, CoBot,' it ignores you," she says. "In the future it's got to respond to spontaneous interaction." And it remains to be seen how people will react when the sight of an autonomous robot in the hallways is no longer a novelty, but an everyday occurrence. "Inevitably, as it's moving around and talking, will people get annoyed and yell at it, or will they be happy at the sound of its voice?" Veloso says. "All of these problems are things we need to study. These are the kinds of questions that interest me."

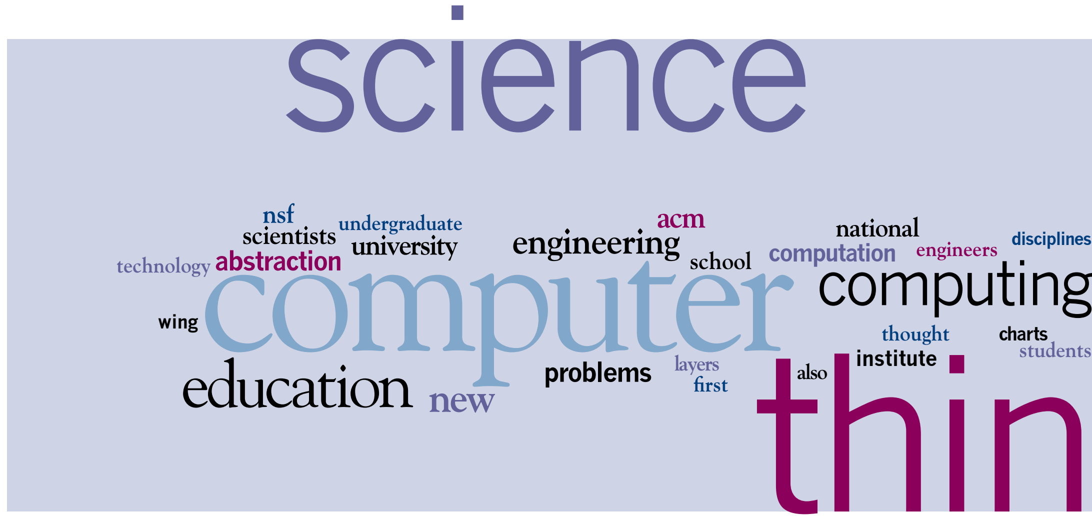
> "The idea is that if we can personalize the interaction, will that make a difference in how people interact with the robot?'
>
> *Reid Simmons*

It may have taken a lot longer than 1950s futurists imagined, but social robots are likely to become ubiquitous in people's lives, says Simmons, especially in providing assistance to the elderly or disabled. "They will become the most complicated technology that people will interact with, and they'll be operated by novices—people who don't have training in robotics," he says. "And my feeling is that we can either make the technology something that's easy to learn, or we can make it something they're familiar with and don't have to learn."

Robots suitable for home use are still limited in their ability to navigate autonomously and manipulate objects, Simmons says, but those capabilities are steadily improving, and researchers need to be pushing the development of social interfaces at the same rate. "My hope is that when the manipulation and mobility technologies are ready, the social interfaces will be ready," he says.

Computational Thinking—What and Why?

By Jeannette M. Wing



In an March 2006 article for the Communications of the ACM, I used the term "computational thinking" to articulate a vision that everyone, not just those who major in computer science, can benefit from thinking like a computer scientist [Wing06]. So, what is computational thinking? Here's a definition that Jan Cuny of the National Science Foundation, Larry Snyder of the University of Washington and I use; it was inspired by an email exchange I had with Al Aho of Columbia University:

> *Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent.* [CunySnyderWing10]

Informally, computational thinking describes the mental activity in formulating a problem to admit a computational solution. The solution can be carried out by a human or machine, or more generally, by combinations of humans and machines.

My interpretation of the words "problem" and "solution" is broad. I mean not just mathematically well-defined problems whose solutions are completely analyzable, e.g., a proof, an algorithm or a program, but also real-world problems whose solutions might be in the form of large, complex software systems. Thus, computational thinking overlaps with logical thinking and systems thinking. It includes algorithmic thinking and parallel thinking, which in turn engage other kinds of thought processes, such as compositional reasoning, pattern matching, procedural thinking and recursive thinking. Computational thinking is used in the design and analysis of problems and their solutions, broadly interpreted.

The Value of Abstraction

The most important and high-level thought process in computational thinking is the abstraction process. Abstraction is used in defining patterns, generalizing from specific instances and parameterization. It is used to let one object stand for many. It is used to capture essential properties common to a set of objects while hiding irrelevant distinctions among them. For example, an algorithm is an abstraction of a process that takes inputs, executes a sequence of steps and produces outputs to satisfy a desired goal. An abstract data type defines an abstract set of values and operations for manipulating those values, hiding the actual representation of the values from the user of the abstract data type. Designing efficient algorithms inherently involves designing abstract data types.

Abstraction gives us the power to scale and deal with complexity. Applying abstraction recursively allows us to build larger and larger systems, with the base case (at least for computer science) being bits (0s and 1s). In computing, we routinely build systems in terms of layers of abstraction, allowing us to focus on one layer at a time and on the formal relations (e.g., "uses," "refines" or "implements," "simulates") between adjacent layers. When we write a program in a high-level language, we're building on lower layers of abstractions. We don't worry about the details of the underlying hardware, the operating system, the file system or the network; furthermore, we rely on the compiler to correctly implement the semantics of the language. The narrow-waist architecture of the Internet demonstrates the effectiveness and robustness of appropriately designed abstractions: the simple TCP/IP layer at the middle has enabled a multitude of unforeseen applications to proliferate at layers above, and a multitude of unforeseen platforms, communications media and devices to proliferate at layers below.

Computational thinking draws on both mathematical thinking and engineering thinking. Unlike mathematics, however, our computing systems are constrained by the physics of the underlying information-processing agent and its operating environment. And so, we must worry about boundary conditions, failures, malicious agents and the unpredictability of the real world. And unlike other engineering disciplines, in computing —thanks to software, our unique "secret weapon"—we can build virtual worlds that are unconstrained by physical realities. And so, in cyberspace our creativity is limited only by our imagination.

Computational Thinking and Other Disciplines

Computational thinking has already influenced the research agenda of all science and engineering disciplines. Starting decades ago with the use of computational modeling and simulation, through today's use of data mining and machine learning to analyze massive amounts of data, computation is recognized as the third pillar of science, along with theory and experimentation [PITAC 2005].

The expedited sequencing of the human genome through the "shotgun algorithm" awakened the interest of the biology community in computational methods, not just computational artifacts (such as computers and networks). The volume and rate at which scientists and engineers are

computational

teachers just data course programs used microsoft research schools use cra google mathematics systems everyone solutions

king

now collecting and producing data—through instruments, experiments and simulations—are demanding advances in data analytics, data storage and retrieval, as well as data visualization. The complexity of the multi-dimensional systems that scientists and engineers want to model and analyze requires new computational abstractions. These are just two reasons that every scientific directorate and office at the National Science Foundation participates in the Cyber-enabled Discovery and Innovation, or CDI, program, an initiative started four years ago with a fiscal year 2011 budget request of $100 million. CDI is in a nutshell "computational thinking for science and engineering."

Computational thinking has also begun to influence disciplines and professions beyond science and engineering. For example, areas of active study include algorithmic medicine, computational archaeology, computational economics, computational finance, computation and journalism, computational law, computational social science and digital humanities. Data analytics is used in training army recruits, detecting email spam and credit card fraud, recommending and ranking the quality of services and even personalizing coupons at supermarket checkouts.

At Carnegie Mellon, computational thinking is everywhere. We have degree programs, minors, or tracks in "computational X" where X is applied mathematics, biology, chemistry, design, economics, finance, linguistics, mechanics, neuroscience, physics and statistical learning. We even have a course in computational photography. We have programs in computer music, and in computation, organizations and society. The structure of our School of Computer Science hints at some of the ways that computational thinking can be brought to bear on other disciplines. The Robotics Institute brings together computer science, electrical engineering and mechanical engineering; the Language Technologies Institute, computer science and linguistics; the Human-Computer Interaction Institute, computer science, design, and psychology; the Machine Learning Department, computer science and statistics; the Institute for Software Research, computer science, public policy and social science. The newest kid on the block, the Lane Center for Computational Biology, brings together computer science and biology. The Entertainment Technology Center is a joint effort of SCS and the School of Drama. SCS additionally offers joint programs in algorithms, combinatorics and optimization (computer science, mathematics and business); computer science and fine arts; logic and computation (computer science and philosophy); and pure and applied logic (computer science, mathematics and philosophy).

Computational Thinking in Daily Life

Can we apply computational thinking in daily life? Yes! These stories helpfully provided by Computer Science Department faculty demonstrate a few ways:

Pipelining: SCS Dean Randy Bryant was pondering how to make the diploma ceremony at commencement go faster. By careful placement of where individuals stood, he designed an efficient pipeline so that upon the reading of each graduate's name and honors by Assistant Dean Mark Stehlik, each person could receive his or her diploma, then get a handshake or hug from Mark and then get his or her picture taken. This pipeline allowed a steady stream of students to march across the stage (though a pipeline stall occurred whenever the graduate's cap would topple while getting a hug from Mark).

Seth Goldstein, associate professor of computer science, once remarked to me that most buffet lines could benefit from computational thinking: "Why do they always put the dressing before the salad? The sauce before the main dish? The silverware at the start? They need some pipeline theory."

Hashing: After giving a talk at a department meeting about computational thinking, Professor Danny Sleator told me about a hashing function his children use to store away Lego blocks at home. According to Danny, they hash on several different categories: rectangular thick blocks, other thick (non-rectangular) blocks, thins (of any shape), wedgies, axles, rivets and spacers, "fits on axle," ball and socket and "miscellaneous." They even have rules to classify pieces that could fit into more than >>>

one category. "Even though this is pretty crude, it saves about a factor of 10 when looking for a piece," Danny says. Professor Avrim Blum overheard my conversation with Danny and chimed in "At our home, we use a different hash function."

Sorting: The following story is taken verbatim from an email sent by Roger Dannenberg, associate research professor of computer science and professional trumpeter. "I showed up to a big band gig, and the band leader passed out books with maybe 200 unordered charts and a set list with about 40 titles we were supposed to get out and place in order, ready to play. Everyone else started searching through the stack, pulling out charts one-at-a-time. I decided to sort the 200 charts alphabetically O(N log(N)) and then pull the charts O(M log(N)). I was still sorting when other band members were halfway through their charts, and I started to get some funny looks, but in the end, I finished first. That's computational thinking."

Benefits of Computational Thinking

Computational thinking enables you to bend computation to your needs. It is becoming the new literacy of the 21st century. Why should everyone learn a little computational thinking? Cuny, Snyder and I advocate these benefits [CunySnyderWing10]:

Computational thinking for everyone means being able to:

- Understand which aspects of a problem are amenable to computation,
- Evaluate the match between computational tools and techniques and a problem,
- Understand the limitations and power of computational tools and techniques,
- Apply or adapt a computational tool or technique to a new use,
- Recognize an opportunity to use computation in a new way, and
- Apply computational strategies such divide and conquer in any domain.

Computational thinking for scientists, engineers and other professionals further means being able to:

- Apply new computational methods to their problems,
- Reformulate problems to be amenable to computational strategies,
- Discover new science through analysis of large data,
- Ask new questions that were not thought of or dared to ask because of scale, but which are easily addressed computationally, and
- Explain problems and solutions in computational terms.

Computational Thinking in Education

Campuses throughout the United States and abroad are revisiting their undergraduate curriculum in computer science. Many are changing their first course in computer science to cover fundamental principles and concepts, not just programming. For example, at Carnegie Mellon we recently revised our undergraduate first-year courses to promote computational thinking for non-majors [Link10].

Moreover, the interest and excitement surrounding computational thinking has grown beyond undergraduate education to additional recent projects, many focused on incorporating computational thinking into kindergarten through 12th grade education. Sponsors include professional organizations, government, academia and industry.

The College Board, with support from NSF, is designing a new Advanced Placement (AP) course that covers the fundamental concepts of computing and computational thinking (see the website www.csprinciples.org). Five universities are piloting versions of this course this year: University of North Carolina at Charlotte, University of California at Berkeley, Metropolitan State College of Denver, University of California at San Diego and University of Washington. The plan is for more schools—high schools, community colleges and universities—to participate next year.

Computer science is also getting attention from elected officials. In May 2009, computer science thought leaders held an event on Capitol Hill to call on policymakers to put the "C" in STEM, that is, to make sure that computer science is included in all federally funded educational programs that focus on science, technology, engineering and mathematics (STEM) fields. The event was sponsored by ACM, CRA, CSTA, IEEE, Microsoft, NCWIT, NSF and SWE. The U.S. House of Representatives has now designated the first week of December as Computer Science Education Week (www.csedweek.org); the event is sponsored by ABI, ACM, BHEF, CRA, CSTA, Dot Diva, Google, Globaloria, Intel, Microsoft, NCWIT, NSF, SAS and Upsilon Pi Epsilon. In July 2010, U.S. Rep. Jared Polis (D-CO) introduced the Computer Science Education Act (H.R. 5929) in an attempt to boost K-12 computer science education efforts.

Another boost is expected to come from the NSF's Computing Education for the 21st Century (CE21) program, started in September 2010 and designed to help K-12 students, as well as first- and second-year college students, and their teachers develop computational thinking competencies. CE21 builds on the successes of the two NSF programs, CISE Pathways to Revitalized Undergraduate Computing Education (CPATH) and Broadening Participating in Computing (BPC). CE21 has a special emphasis on activities that support the CS 10K Project, an initiative launched by NSF through BPC. CS 10K aims to catalyze a revision of high school curriculum, with the proposed new AP course as a centerpiece, and to prepare 10,000 teachers to teach the new courses in 10,000 high schools by 2015.

Industry has also helped promote the vision of computing for all. Since 2006, with help from Google and later Microsoft, Carnegie Mellon has held summer workshops for high school teachers called "CS4HS." Those workshops are designed to deliver the message that there is more to computer science than computer programming. CS4HS spread in 2007 to UCLA and the University of Washington. By 2010, under the auspices of Google, CS4HS had spread to 20 schools in the United States and 14 in Europe, the Middle East and Africa. Also at Carnegie Mellon, Microsoft Research funds the Center for Computational Thinking (www.cs.cmu.edu/~CompThink/), which supports both research and educational outreach projects.

Computational thinking has also spread internationally. In August 2010, the Royal Society—the U.K.'s equivalent of the U.S.'s National Academy of Sciences—announced that it is leading an 18-month project to look "at the way that computing is taught in schools, with support from 24 organizations from across the computing community including learned societies, professional bodies, universities and industry." (See www.royalsociety.org/education-policy/projects/.) One organization that has already taken up the challenge in the U.K. is called Computing At School, a coalition run by the British Computing Society and supported by Microsoft Research and other industry partners.

Resources Abound

The growing worldwide focus on computational thinking means that resources are becoming available for educators, parents, students and everyone else interested in the topic.

In October 2010, Google launched the Exploring Computational Thinking website (www.google.com/edu/computational-thinking), which has a wealth of links to further web resources, including lesson plans for K-12 teachers in science and mathematics.

Computer Science Unplugged (www.csunplugged.org), created by Tim Bell, Mike Fellows and Ian Witten, teaches computer science without the use of a computer. It is especially appropriate for elementary and middle school children. Several dozen people working in many countries, including New Zealand, Sweden, Australia, China, Korea, Taiwan and Canada, as well as in the United States, contribute to this extremely popular website.

The National Academies' Computer Science and Telecommunications Board held a series of workshops on "Computational Thinking for Everyone" with a focus on identifying the fundamental concepts of computer science that can be taught to K-12 students. The first workshop report [NRC10] provides multiple perspectives on computational thinking.

Additionally, panels and discussions on computational thinking have been plentiful at venues such as the annual ACM Special Interest Group on Computer Science Education (SIGCSE) symposium and the ACM Educational Council. The education committee of the CRA presented a white paper [CRA-E10] at the July 2010 CRA Snowbird conference, which includes recommendations for computational thinking courses for non-majors. CSTA produced and distributes "Computational Thinking Resource Set: A Problem-Solving Tool for Every Classroom." It's available for download at the CSTA's website (www.csta.acm.org).

Final Remarks—and a Challenge

Computational thinking is not just or all about computer science. The educational benefits of being able to think computationally—starting with the use of abstractions—enhance and reinforce intellectual skills, and thus can be transferred to any domain.

Computer scientists already know the value of thinking abstractly, thinking at multiple levels of abstraction, abstracting to manage complexity, abstracting to scale up, etc. Our immediate task ahead is to better explain to non-computer scientists what we mean by computational thinking and the benefits of being able to think computationally. Please join me in helping to spread the word!

Jeannette Wing is head of the Computer Science Department at Carnegie Mellon University and the President's Professor of Computer Science. She earned her bachelor's, master's and doctoral degrees at the Massachusetts Institute of Technology and has been a member of the Carnegie Mellon faculty since 1985.

From 2007 to 2010, Wing served as assistant director for the Computer and Information Science and Engineering Directorate of the National Science Foundation. She is a fellow of the American Academy of Arts and Sciences, the American Association for the Advancement of Science, the Association for Computing Machinery and the Institute of Electrical and Electronic Engineers.

Bibliography and Further Reading

[CRA-E10] Computer Research Association, "Creating Environments for Computational Researcher Education," August 9, 2010 www.cra.org/uploads/documents/resources/rissues/CRA-E-Researcher-Education.pdf

[CunySnyderWing10] Jan Cuny, Larry Snyder and Jeannette M. Wing, "Demystifying Computational Thinking for Non-Computer Scientists," work in progress, 2010

[NRC10] "Report of a Workshop on the Scope and Nature of Computational Thinking," National Research Council, 2010 (www8.nationalacademies.org/cp/projectview.aspx?key=48969)

[Link10] Randal E. Bryant, Klaus Sutner and Mark Stehlik, "Introductory Computer Science Education: A Dean's Perspective," The Link, Fall 2010

[PITAC05] President's Information Technology Advisory Council, "Computational Science: Ensuring America's Competitiveness," Report to the President, June 2005

[Wing06] Jeannette M. Wing, "Computational Thinking," Communications of the ACM, Vol. 49, No. 3, March 2006, pp. 33–35

Acronyms of Organizations:

ABI: Anita Borg Institute for Women and Technology

ACM: Association for Computing Machinery

BHEF: Business-Higher Education Forum

CISE: Computer and Information Science and Engineering

CRA: Computing Research Association

CRA-E: Computing Research Association-Education

CSTA: Computer Science Teachers Association

CSTB: Computer Science and Telecommunications Board

IEEE: Institute for Electrical and Electronic Engineers

NCWIT: National Center for Women and Information Technology

NSF: National Science Foundation

SIGCSE: ACM Special Interest Group on Computer Science Education

SWE: Society for Women Engineers

Acronyms of NSF Programs:

BPC: Broadening Participating in Computing

CDI: Cyber-enabled Discovery and Innovation

CE21: Computing Education for the 21st Century

CPATH: CISE Pathways to Revitalized Undergraduate Computing Education

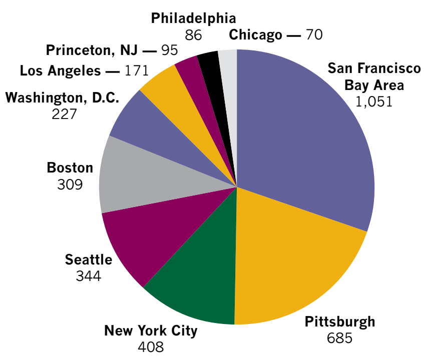
Tina M. Carr (HNZ'02)
Director of Alumni Relations
School of Computer Science
tcarr@cs.cmu.edu

SCS Alumni At-A-Glance*

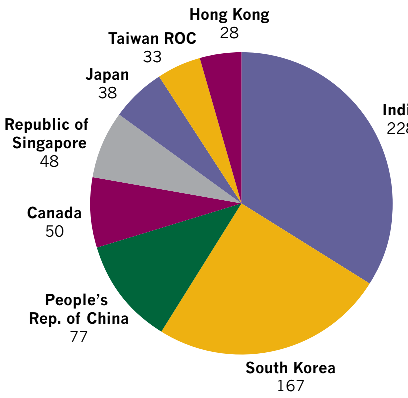
Total Alumni: 5,675
Male: 4,606
Female: 1,069

ALUMNI BY GEOGRAPHY

Domestic Top Ten (by city)



International
Top Eight (by country)



* As of Oct. 2010

Connecting: A Cure for Winter Blahs

Winter doldrums. The blahs. The blues. Let's face it: The weather around here right now does not make me feel very inspired or energetic. In fact, the endless days of snow, cold and gray skies are enough to make a person want to either hibernate like a bear for the rest of the winter or permanently relocate to Bora Bora (I can see the smiles of the year-round temperate climate residing alumni now).

Certainly, neither option is implausible given our ability to stay connected to the world. Many of us could pretty much work from anywhere these days. As long as there was a network connection, my work would continue without interruption.

We now have more ways than ever to help alumni stay connected to SCS—all of which can serve as a distraction to what's going on outside.

In today's social media space, we can communicate with the SCS community using tools like Facebook, Twitter, LinkedIn, YouTube, Flickr and iTunes. At their convenience, alumni learn about the latest research and educational developments, learn about on-campus and regional alumni events, and connect with fellow alumni.

The benefits of incorporating these tools into our alumni relations outreach are innumerable. We are able to reach out to the community and share information not only in a timelier manner, but often as it's happening. In turn, we can receive instant feedback and comments from the community on posted news stories for example.

Social media tools allow us to reach a broader audience—people we might not be reaching through the traditional methods such as email. It also helps foster a greater sense of community among alumni and between alumni and the university.

The opportunities to connect and hear from the SCS community in the virtual world are not only essential, but are a valuable part of our outreach strategies. However, while social media tools are playing a more pivotal role in our goal of increasing alumni engagement, I also still recognize the importance of meeting people face-to-face.

A good portion of my time is spent traveling and meeting with alumni, either one on one or during events. After nearly 11 years, I can say without a doubt that the face-to-face time I spend with alumni is the most rewarding. I've had many alumni tell me, "Oh, you're the one that sends all those emails. It's nice to put a face with a name." I feel the same.

Taking the time to sit down with alumni allows me to get to know them on a friendly level and creates a personal connection. It's a chance for me to learn more about their personal and professional interests and how their interests might be enhanced by volunteer activities. These symbiotic connections are crucial in building meaningful and long-lasting relationships with our alumni.

The more we learn about our individual alumni, the better we are able to develop programs and activities that will fit the needs and interests of the whole community.

Ultimately, we hope by providing alumni more opportunities to hear about the great things happening around the school, you will stay connected and become engaged by our unique opportunities. Using social media outlets and having an open dialogue with our alumni community are keys to building a vibrant, active alumni community.

So if you haven't already, check out the latest on Facebook (SCSatCMU or CarnegieMellonU), follow us on Twitter (@SCSatCMU), or watch a lecture via iTunesU.

Then join us at one of the many upcoming alumni events, like Network Night D.C. or Boston. (You can see our complete calendar at www.cmu.edu/alumni.)

Also, Spring Carnival and Reunion Weekend (April 14–16) are just around the corner! There will be a special joint SCS and ECE alumni reception as well as numerous campus-wide activities including reunions, mobot races, buggy, midway, lectures, concerts and more. We look forward to welcoming everyone back to campus.

Hope to see and hear from you.

Tina M. Carr

Michael Livanos (CS'04), Rajashekar Reddy (CS'04), Jim Bai (CS'05, S'05) and Craig Austin (CS'05)

Connie Chau (CS'04) and Kavin Du (CS'01)

Marc Donner (CS'82,'84), Philip Lehman (CS'78,'84) and Satish Gupta, (CS'79,'82)

SCS and ECE alumni gathered Dec. 11 for the annual New York alumni holiday brunch at the Westin Times Square. Over 70 alumni and friends attended the lively event, with SCS Dean Randy Bryant and ECE Department Head Ed Schlesinger updating the crowd on the latest educational and research developments from their respective areas.

Giving Back

> **By Mark Dorgan**

Daniel Siewiorek has witnessed much in his time at Carnegie Mellon.

A graduate of the University of Michigan and Stanford University, Dan worked with pioneers in both artificial intelligence (Allen Newell, Herb Simon, Raj Reddy) and computer architecture (Gordon Bell) who provided him with a unique and broad perspective on those fields.

Daniel and Karon Siewiorek

"From my first encounters, it was clear to me that CMU was a very special culture—multidisciplinary, cooperative, and they built systems that made a difference—all guided by 'the reasonable person principle'—an ideal, nurturing, 'can-do' environment to grow into a mature researcher," says Dan, CMU's Buhl University Professor of Computer Science and Electrical and Computer Engineering.

Dan epitomizes the interdisciplinary approach for which the university is noted, and over a span of four decades at CMU has been involved in leading teams that have designed and constructed over 20 mobile computing systems. Among his seminal contributions was work on the Cm* project that culminated in an operational 50-processor multiprocessor system. He is also a key contributor to the dependability design of more than two-dozen commercial computing systems. Most recently, he served as head of the Human-Computer Interaction Institute in the School of Computer Science.

As the Gates Center for Computer Science was being built, he and his wife Karon recognized the significance and importance the building marked in the history of SCS. He and Karon made a gift to support the Siewiorek-Walker Classroom on the fifth floor of the Gates Center.

"Our parents understood the importance of education and gave us opportunities they themselves never had," Dan says. "We chose to support a classroom to honor their commitment to us. In addition we have known the joy of working with students and thriving by sharing in their enthusiasm. We picked the classroom adjacent to the Pausch Bridge to commemorate CMU's multi-disciplinary culture."

Dan and Karon's gifts were made as part of Carnegie Mellon's "Inspire Innovation" campaign, which has raised nearly $688 million as of Jan. 1. To find how you can help the School of Computer Science through scholarships, fellowships, faculty support or gifts toward the Gates and Hillman Centers, please contact me at mdorgan@cmu.edu or call me at 412-268-8576. You can also learn more about the Inspire Innovation campaign by visiting www.cmu.edu/campaign.

Mark Dorgan is executive director of major gifts and development liaison for the School of Computer Science.

Diana Yu

B.S., industrial management, Carnegie Mellon University, 1999
B.S., information systems, Carnegie Mellon University, 1999
M.H.C.I., human-computer interaction, Carnegie Mellon University, 2008

It might seem like a long way from writing HTML and Javascript to overseeing construction projects on one of the nation's busiest commuter railroads. But for Diana Yu, the journey is one of only a few yards inside the Los Angeles headquarters of Metrolink, which serves six California counties, including Los Angeles and San Diego.

Yu is transitioning from a role in the agency's communications department to becoming an engineering program management analyst. "It's really interesting, and I'm excited," she says. "We're still going through a lot of changes, and it feels almost like a startup in some ways. It's a really good time to be here to contribute."

Metrolink's 512-mile transit system carries more than 38,000 passengers a day. A new CEO was named in 2010, and set as his goals improved safety procedures, expanded service and upgraded communication with riders and the general public.

Social media has provided important tools to spread the news about the new Metrolink, says Yu, who joined the agency two years ago. Working with a tiny budget, Yu conducted a needs assessment using accumulated customer feedback. "Customers wanted service alerts so they would know the status of their trains, they wanted train schedules in specific formats, and they were interested in special offers," she says. Yu says she was able to draw heavily on lessons she learned from Shelley Evenson's Designing for Service class at Carnegie Mellon.

The result was a brand-new website (designed in only three months) especially for mobile phone users, Yu says.

Metrolink also now uses Twitter to provide system updates so that riders don't have to check a website, she says.

Yu's new job utilizes her undergraduate experiences at CMU, where she earned degrees in business and information systems, as well as her eight years as a consultant with IBM Corp. "In general, this role is working with our engineering project managers when they start new projects to ensure that they run within budget, and on schedule," says Yu, who also providing user experience expertise for a complete redesign of Metrolink's website.

In her free time, Yu enjoys canoeing—lately as part of a six-person team that paddles an outrigger canoe in races up and down the Pacific coast. A former member of the CMU crew team, Yu has loved working out on the water since her college summers in Boston, when her sister, then an MIT undergrad, used to take her out on the Charles River. "One of the nicest things about living in southern California is that I'm only 10 or 15 minutes from the ocean," Yu says. —Jason Togyer (HS'96)

Jerry Zhu

B.S., computer science, Shanghai Jiao Tong University, 1993
M.S., computer science, Shanghai Jiao Tong University, 1996
Ph.D., Language Technologies Institute, Carnegie Mellon University, 2005

If any machine-learning research can be considered "retro," that might be an apt description of the work Xiaojin (Jerry) Zhu is pursuing at the University of Wisconsin at Madison.

Zhu, an assistant professor of computer science, is investigating the ways that human cognition can be studied using machine-learning techniques and vice versa. He says his work is almost a throwback to what's now considered "classical" artificial intelligence research as performed by Herbert Simon, Allen Newell and other AI pioneers a half-century ago.

"I'm interested in finding the fundamental mathematical principles that govern learning across the spectrum, using both human subjects and computers," says Zhu, who collaborates closely with UW-Madison's psychology department.

There are strong comparisons to be made in the ways that humans and computers acquire new knowledge, he says. Take the problem of over-fitting. Over-fitting happens when a machine is given a "training set" of data and creates a model too exactly fitted to the data—one that finds not the true underlying pattern, but instead the idiosyncrasy of that particular training set.

"It turns out this is relevant in humans as well," Zhu says. In one experiment at UW-Madison, students were given a list of five words, together with their category. For example, "daylight" was listed as a word in "category A," while the words "hospital," "termite," "envy" and "scream" were in "category B." Students were then asked to predict the category of more words. Zhu says students came up with elaborate explanations (i.e., over-fit) why "daylight" belonged in category A, while the others belonged in B. The actual reason was simple: Category A represented words with positive connotations.

"Of course it is hard to figure out the actual rule with only five words, and easy to come up with wrong guesses," Zhu says. "The real question, however, is whether we can derive a precise mathematical formula on how badly humans will over-fit given any training set." Using a machine-learning concept known as Rademacher complexity, he and his collaborators developed a mathematical model that predicted exactly that. Such models, though highly theoretical, could have applications in education—for instance, in predicting how likely students are to grasp underlying concepts from examples they see in classes or textbooks.

"I hope machine learning will eventually come back to address more of the cognitive science problems that classic AI considered," says Zhu, who jokes that the Machine Learning Department at CMU might then be renamed simply the "Learning Department."

In his spare time, Zhu enjoys amateur astronomy, sometimes looking at the night sky near Madison through his own 8 inch Dobsonian. Zhu, his wife and their children, ages 2 and 6, also participate in family fossil-hunting trips organized by Madison's geology museum.
—Jason Togyer (HS'96)

It's the University Whose Technology Helped Power the Jeopardy!-Winning Computer

What is Carnegie Mellon University?

A standing-room-only crowd filled the Rashid Auditorium on Feb. 16 to view three episodes of Jeopardy! featuring IBM's Watson question-answering system, including the live broadcast of the final episode in which Watson convincingly beat human champions Ken Jennings and Brad Rutter.

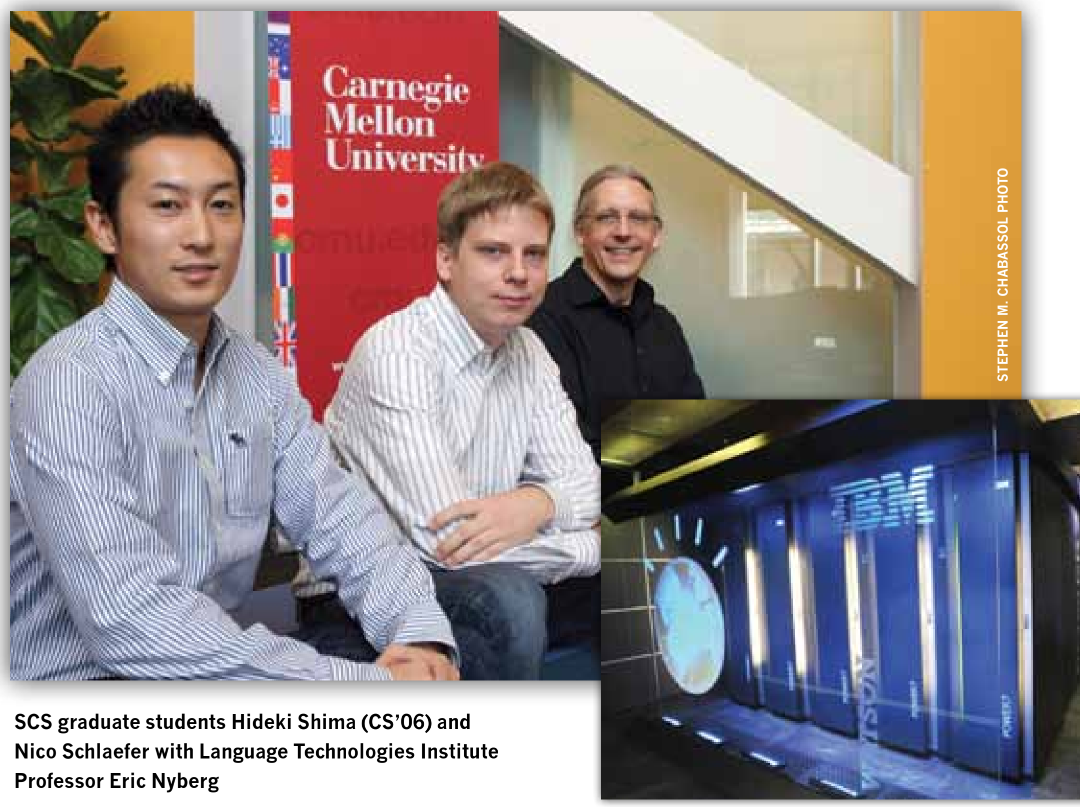
Eric Nyberg, professor in the Language Technologies Institute, told the audience how question-answering research began at Carnegie Mellon, and how LTI faculty and students were involved in IBM's Watson project from its very beginning. Two of his Ph.D. students, Nico Schlaefer and Hideki Shima (CS'06), developed important pieces of Watson's software during internships at IBM. Mark Sherman (CS'79,'83), program director for IBM Software Group Strategy, also spoke prior to the finale.

CMU is one of eight universities that collaborated with IBM on the Watson system. The partnership between IBM and CMU to investigate question-answering technology began in 2001.

Meanwhile, Carnegie Mellon's role in developing the technology behind Watson was the subject of an episode of PBS's top-rated science documentary series, Nova. The episode "The Smartest Machine on Earth," which aired in most of the U.S. on Feb. 9, included commentary from Tom Mitchell, head of the Machine Learning Department, and Luis von Ahn, assistant professor of computer science. Alex Waibel, professor in the Language Technologies Institute, demonstrated his Jibbigo language translation app on the program.

In addition to the full episode, NOVA's website includes shorter features on artificial intelligence, including a video about robot soccer featuring Manuela Veloso, professor of computer science, and another about computer vision featuring Alexei Efros, associate professor of robotics and computer science.

A commentary by Nyberg about the implications of Watson's victory for the question-answering research community has been posted on the Inside Nova website. Nyberg and Tom Mitchell, head of the Machine Learning Department, also live-blogged the Feb. 16 Jeopardy! episode for Nova, while Shima and Schlaefer likewise live-blogged the Feb. 14 episode. All of those blog posts—and the videos—can be found at the Nova website, www.pbs.org/wgbh/nova. (Look for the episode titled, "The Smartest Machine on Earth.")



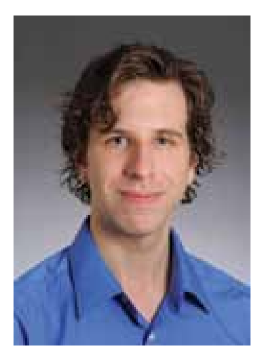
STEPHEN M. CHABASSOL PHOTO

SCS graduate students Hideki Shima (CS'06) and Nico Schlaefer with Language Technologies Institute Professor Eric Nyberg

Platzer, Smith receive NSF Early Career awards

Two SCS faculty members in January received prestigious Faculty Early Career Development Awards from the National Science Foundation.

Noah A. Smith, assistant professor of language technologies, has received a $550,000, five-year award to study flexible statistical learning algorithms for natural language processing; while Andre Platzer, assistant professor of computer science, has received a $400,000, five-year award to study the logical foundations of hybrid computer-physical systems, the NSF announced.



Noah A. Smith

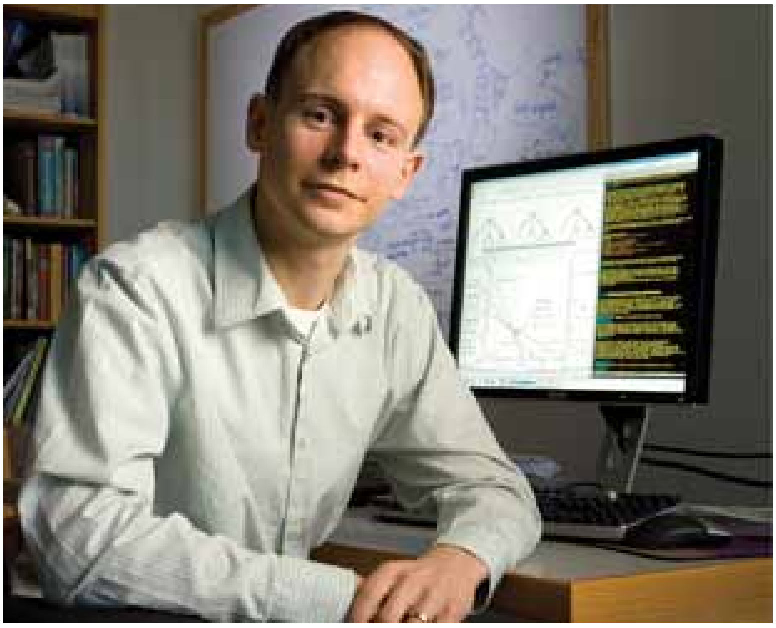
Andre Platzer

According to the NSF, the Faculty Early Career Development program offers the agency's most prestigious awards in support of junior faculty who "exemplify the role of teacher-scholars" through the integration of "outstanding research" with "excellent education." Since the program's inception, 32 members of the Carnegie Mellon faculty have received early-career development awards from the NSF—more than any other Pennsylvania university.

Smith came to the School of Computer Science in 2006. A graduate of the University of Maryland and Johns Hopkins, his research focuses on computational models of human language: formal aspects, learning such models from data and applying them to problems such as translation and social media analysis.

Platzer joined SCS in 2008. A graduate of the University of Karlsruhe and the University of Oldenburg, his research focuses on developing methods for verifying the performance of so-called "cyber-physical" systems, in which embedded computers interact with the real world, and where mistakes or malfunctions can jeopardize safety.

SCS alum to receive top AI honor in July



Vincent Conitzer

An SCS alumnus will receive one of the top awards for researchers in artificial intelligence at a conference in Barcelona this July.

Vincent Conitzer (CS'03,'06) has been named the 2011 recipient of the Computers and Thought Award from the International Joint Conference on Artificial Intelligence. The award is presented every two years to the world's leading AI researchers under the age of 35.

An assistant professor of computer science at Duke University, Conitzer earned his master's and Ph.D. degrees at Carnegie Mellon. The award recognizes his groundbreaking research on computational aspects of game theory, social choice and mechanism design.

Conitzer is the first "third-generation" winner of an IJCAI award—his doctoral adviser at Carnegie Mellon was Tuomas Sandholm, who received the Computers and Thought Award in 2003; while Sandholm's doctoral adviser was Victor Lesser, who received the IJCAI Award for Research Excellence in 2009.

Researchers: Creative work can be crowd-sourced, too

Writing is usually depicted as a solitary intellectual pursuit, but a researcher at CMU's Human-Computer Interaction Institute says writing an article also can be accomplished by dozens of people working independently online.

The research team led by Niki Kittur, assistant professor of human-computer interaction, found that the crowd-sourced articles compared favorably with articles written by a single author and with entries in the Simple English Wikipedia.

Along with Robert Kraut, professor of human-computer interaction, and Boris Smus, a student in HCII's joint master's degree program with the University of Madeira, Kittur approached the crowd-sourcing market as if it was a distributed computing system, such as those used for Web searches.

The researchers created a framework called CrowdForge that breaks down complex tasks into simple, independent micro-tasks that can be completed rapidly and cheaply. Each person in the experiments completed just a sliver of the work of preparing an article, such as preparing an outline, gathering facts or assembling facts into simple prose. The "authors" never even spoke with each other.

"This is exciting because collaborative crowd-sourcing could change the future of work," Kittur says. "We foresee a day when it will be possible to tap into hundreds of thousands or millions of workers around the globe to accomplish creative work on an unprecedented scale."

Two San Francisco-based science journalists, have now created a blog called "My Boss is a Robot" to explore the use of CrowdForge for preparing science news articles based on research reports. Kittur's work was supported in part by grants from the National Science Foundation.

HCII Ph.D. student wins $50K grand prize

Derek Lomas

Derek Lomas, a Ph.D. student in the Human-Computer Interaction Institute, won the $50,000 grand prize in the $100K Challenge, a competition sponsored by Marvell Technology Group to inspire and reward innovative new educational apps for tablet computers. Marvell announced the prize in January at the Consumer Electronics Show in Las Vegas.

Lomas' winning entry was Battleship Numberline, an educational game that helps strengthen math skills. Lomas is a founder of PlayPower, which received a $180,000 MacArthur Foundation grant in 2009 to develop educational games that can be played on inexpensive computers available in many developing nations. PlayPower is also the subject of a Project Olympus probe that's developing a revenue model to support continued research and development.

Based in Santa Clara, Calif., Marvell produces semiconductors and storage devices for consumer electronics products.

Under the Sea

The two-story Open Oceans tank at the Pittsburgh Zoo & PPG Aquarium now contains 100,000 gallons of salt water, 30 species of sea life—and one submersible robot, or "reefbot."

Young visitors to the exhibit use a control station to remotely pilot "CLEO" around the tank and use its high-definition video camera to track fish and snap photos. By comparing the images from CLEO with reference photos, visitors can identify the type of fish.

In the process, the young explorers are helping researchers at CMU's Robotics Institute develop software that might someday be used by scientists to automatically detect, classify and count fish in natural habitats.

A joint project of the zoo and the Robotics Institute, CLEO stands for Children Learning through Education and Observation and was funded in part by Spark, a program of Pittsburgh's Sprout Fund. The reefbot was adapted from a commercially available submersible. The robot made its public debut in December.

Ashley Kidd, an aquarist at the zoo, developed the idea along with Justine Kasznica, a local business consultant for high-tech start-ups. David Wettergreen, associate research professor of robotics, oversaw the project at the Robotics Institute, where Ph.D. students Mark Desnoyer, Michael Furlong and Scott Moreland and senior research engineer John Thornton built the robot and developed the software.

The robot's interface and an accompanying website—reefbot.com—were designed by graduate students who took a Fall 2010 course taught by Bonnie John, professor of human-computer interaction.

About a foot and a half long, CLEO moves too slowly to chase fish or cause any damage to itself, the tank or the sea life, Moreland says. Software includes safeguards to prevent the tethered submarine from getting caught in crevices, caves and obstructions.

Desnoyer, whose doctoral thesis will focus on intelligent camera systems, led development of CLEO's smart camera technology, which helps detect fish and may eventually be able to automatically classify fish. Aquarium visitors who use CLEO to identify fish in the tank are helping to train the system.

Though humans are identifying the fish based on photos, what CLEO is learning in this process is a set of attributes that it can associate with particular species, Wettergreen says.

Scientists who study deep coral reefs might be particularly interested in the technologies being developed for CLEO, he says. In contrast to corals that flourish in shallow, tropical waters, deep reefs are difficult for human divers to study in detail. Submersible robots that could identify and count the creatures and organisms on those deep corals would provide invaluable data, Wettergreen says.

—Byron Spice

PHOTOS BY PAUL A. SELVAGGIO, PITTSBURGH ZOO & PPG AQUARIUM

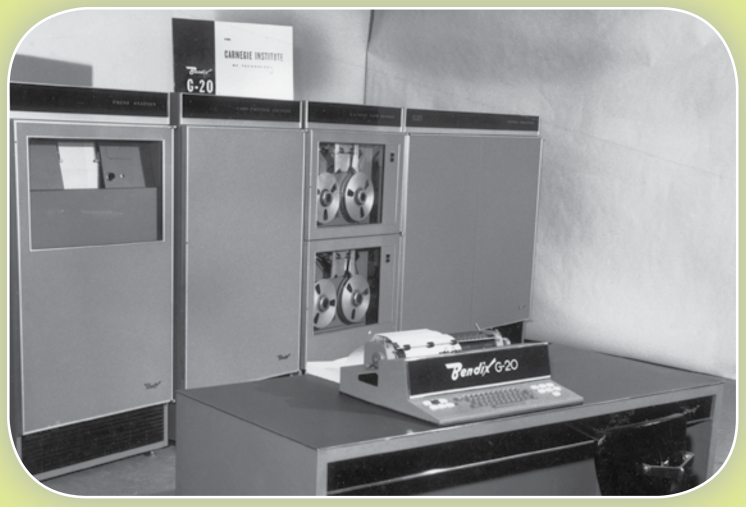
Carnegie Mellon University
School of Computer Science

Office of the Dean
5000 Forbes Avenue
Pittsburgh, PA 15213

Then and Now

There were high hopes for the Bendix G-20 when it arrived at Carnegie Tech 50 years ago this May. Installed in the basement of GSIA, the half-million-dollar machine was the first example of a new series of computers from Bendix Corp., a company better known for making military avionics—and car and truck brakes.

Alan Perlis (S'42), then director of Tech's Computation Center, told The Pittsburgh Press the G-20 would open new frontiers in the study of what he called "machine-man" systems, such as air-traffic control. "There may be research in many other fields—business, the military, early warning or any other systems which need study and improvement," the newspaper said.



UNIVERSITY ARCHIVES

But as Jesse Quatse (S'58, E'62, E'69) remembered in 2006, the G-20's delivery was "a trifle early." The computer didn't yet have "an operating system, compiler or symbolic assembler," he said. "All of the code was translated by hand to binary strings and booted from the punched card transport."

So Perlis gave Arthur Evans Jr. (S'57, '59, '66), Harold Van Zoeren (S'55) and visiting student Jørn Jensen the task of porting an operating system to the G-20 from the university's existing IBM 650. They had the software running in less than one man-year, and Perlis pronounced their work "superb."

Still, the G-20 itself was a developmental dead-end. Bendix Corp. sold fewer than two-dozen before exiting the computer business in 1963. Instead, the G-20 was reused as the heart of something much more important—Carnegie Tech's G-21, one of the first dual-processor computers ever constructed.

For seven years, the one-of-a-kind G-21 on the fourth floor of Scaife Hall served as Carnegie Tech's main research computer, spawned an untold number of projects and spurred creation of the Computer Science Department in 1965. You can learn more about the early days of computing at Carnegie Tech by visiting the marvelous website maintained by Mark DiVecchio (E'70) at www.silogic.com, where you'll also find memories of other CMU machines from days gone by.

For more current perspectives on "machine-man" systems, our report on social robots begins on page 15.

—Jason Togyer (HS'96)