COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

01/09 VOL.52 NO.01

Engineering Rural Development

The Long Road to 64 Bits

Computational Challenges in E-Commerce

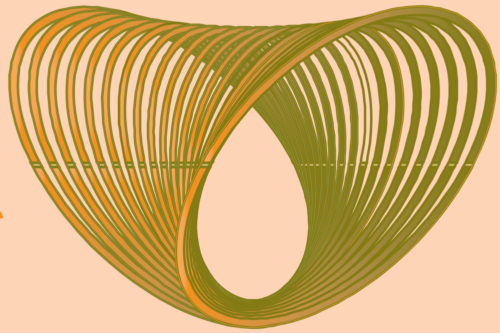
The Legacy of Bill Gates

ACM's FY08 Annual Report

Association for Computing Machinery

2009 SIAM/ACM Joint Conference on

GEOMETRIC & PHYSICAL MODELING



October 5–8, 2009
San Francisco, California
Hilton San Francisco Financial District

IMPORTANT DEADLINES

- Due date for peer-reviewed abstracts for published proceedings .... **March 1, 2009**
- Due date for peer-reviewed technical papers for published proceedings .... **March 16, 2009**
- Due date for minisymposium proposals .... **May 15, 2009**
- Due date for abstracts of all contributed and minisymposium presentations .... **May 15, 2009**

ORGANIZING COMMITTEE

General Chairs

Daniel Gonsor, The Boeing Company
William Regli, Drexel University

General Co-Chairs

Thomas Grandine, The Boeing Company
Jan Vandenbrande, The Boeing Company

Honorary Chairs

George Allen, Siemens PLM Software
Michael Wozny, Rensselaer Polytechnic Institute

Program Chairs

Wim Bronsvoort, Delft University of Technology, The Netherlands
Jens Gravesen, Technical University of Denmark, Denmark
John Keyser, Texas A&M University

Instructions on how to participate and more at

www.siam.org/meetings/gdspm09/

THEMES

Applications

- Computer-aided design, engineering, and manufacturing
- Robotics
- Computer graphics, visualization, and animation
- Virtual environments and prototypes
- Computer vision and image processing
- Biomedical and biochemical applications
- Geophysical applications
- Digital entertainment applications

Topics

- Algebraic and differential geometry
- Computational geometry and topology
- Curves and surfaces
- Geometric and topological representations
- Heterogeneous models for physical objects and properties
- Geometry generation, processing, compression, and transmission
- Reconstruction of surfaces and solids from discrete data
- Shape modeling, synthesis, and analysis
- Geometric constraint solving
- Physics-based modeling
- Conceptual design
- Product and assembly modeling
- Feature modeling and recognition
- Dimensioning and tolerancing
- Simulation and optimization
- Product data exchange, standards, and interoperability
- Collaborative and distributed design
- Haptic and other user interfaces for 3D design

siam
3600 Market Street, 6th Floor
Philadelphia, PA 19104-2688 USA
www.siam.org

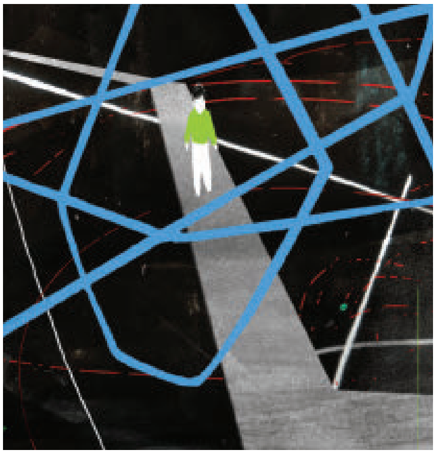
acm
Association for Computing Machinery

Departments

Last Byte

News

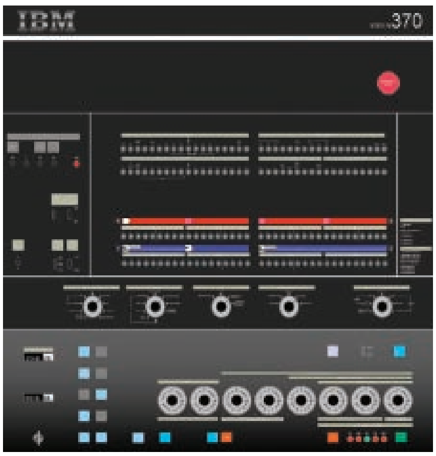
Viewpoints

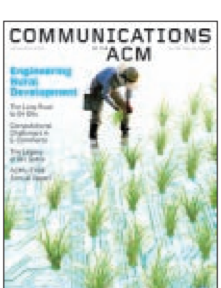


PHOTOGRAPH COURTESY OF THE COMPUTER HISTORY MUSEUM. ILLUSTRATION BY JON HAN

Association for Computing Machinery
Advancing Computing as a Science & Profession

Practice





PHOTOGRAPH BY SELBST GEZEICHNET

About the Cover:
Nongovernmental organizations serve to drive the proliferation and adoption of ICT tools in the rural developing world. Cover art by Alicia Kubista.

Contributed Articles

Review Articles

Research Highlights

Virtual Extension

As with all magazines, page limitations often prevent the publication of articles that might otherwise be included in the print edition. To ensure timely publication, ACM created *Communications*' Virtual Extension (VE).

VE articles undergo the same rigorous review process as those in the print edition and are accepted for publication on their merit. These articles are now available to ACM members in the Digital Library.

COMMUNICATIONS OF THE ACM

A monthly publication of ACM Media

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

ACM U.S. Public Policy Office
Cameron Wilson, Director
1100 Seventeenth St., NW, Suite 507
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Chris Stephenson
Executive Director
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (800) 401-1799; F (541) 687-1840

Association for Computing Machinery (ACM)
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA
T (212) 869-7440; F (212) 869-0481

CONTACT POINTS

Copyright permission
permissions@cacm.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmcoa@cacm.acm.org
Letters to the Editor
letters@cacm.acm.org

WEB SITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/guidelines

BPA Audit Pending

Subscriptions
Annual subscription cost is included in the society member dues of $99.00 (for students, cost is included in $42.00 dues); the nonmember annual subscription rate is $100.00.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM

(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER

Please send address changes to
Communications of the ACM
2 Penn Plaza, Suite 701
New York, NY 10121-0701 USA

Association for Computing Machinery

Printed in the U.S.A.

DOI:10.1145/1435417.1435418

Moshe Y. Vardi

"How Are We Doing?"

A rabbinical story tells about an angry reader who stormed into a newspaper office waving the day's paper, asking to see the editor of the obituary column.

He showed him his name in the obituary listing. "You see," he said, "I am very much alive. I demand a retraction!" The editor replied, "I never retract a story. But I'll tell you what I'll do; I'll put you in the birth announcements and give you a fresh start!"

Communications has had a fresh start; indeed, you hold the seventh issue of the "new CACM." By now, readers should have a fairly solid idea of the new editorial format of *Communications*. When Ed Koch was mayor of New York City, he was famous for asking, "How am I doing?" It is time for *Communications* to ask its membership: "How are we doing?"

Over the last few months we have received hundreds of messages from readers offering us their feedback, which was overwhelmingly, though not universally, positive. Over the coming months, however, we will embark on much more detailed reader surveys, trying to get an in-depth sense of what readers like, or do not like, about the current format of this publication.

In previous letters, I discussed the editorial content model in terms of the new sections we've created and how the articles fit within. Here, I'd like to offer you a peek behind the scenes, and explain how this content is generated. As you know, *Communications* is divided into several sections. Analogously, our Editorial Board is divided into several sections, each operating in a somewhat distinct fashion. (See http://cacm.acm.org/communications?pageIndex=5 for a full listing of the Editorial Board.) Readers should also consult http://cacm.acm.org/guidelines/cacm-author-guidelines/ for detailed author guidelines.

Our News section board is co-chaired by Marc Najork and Prabhakar Raghavan, and supported by Senior Editor/News Jack Rosenberger. This team holds monthly brainstorming teleconferences in which they discuss story ideas. They select three stories for each issue, one with a science focus, one with a technology focus, and one with a societal-impact focus. For each story they identify an expert who can serve as initial source for the writer who will ultimately write the story. Jack, who has built up a network of freelance writers, then identifies a writer for each story and puts the writer in contact with the expert. *Communications* welcomes ideas for news stories; please contact news@cacm.acm.org.

The Viewpoints section board is co-chaired by Susanne Hambrusch, John Leslie King, and J Strother Moore, and supported by Executive Editor Diane Crawford and Managing Editor Thomas Lambert. The Viewpoints section is a forum for the expression of opinion and analysis of a vast range of computing topics, typically of a non-technical nature, and features a combination of regular columns with contributed (solicited and unsolicited) short opinion articles, as well as editorial debates in a point/counterpoint format. This section is intentionally less "scholarly" than the other sections of the magazine, reflecting opinions and viewpoints that are not always backed by scholarly evidence. Regular Viewpoints columns are handled by section board members, who solicit writers who are experts in their fields. Several of these columnists appear regularly; other columns are shared by different writers. Contributed opinion essays (both solicited and unsolicited) are subject to peer review. The co-chairs do the first round of filtering, selecting for further review only those that are well written, focus on topics of a very broad interest, and offer sound arguments. Selected articles are then assigned to a section board member, who oversees the review process.

The last few pages of each issue of *Communications* consist of the Last Byte section, which is overseen by Senior Editors Andrew Rosenbloom and Jack Rosenberger. This section alternates content between the Puzzled column, where Peter Winkler presents mathematic brainteasers; Future Tense, where science fiction writers contribute thought-provoking essays; and Q&A, where Leah Hoffman offers brief interviews with computing personalities.

In my next letter, I will describe the editorial model of the four technical sections of *Communications:* Practice, Contributed Articles, Review Articles, and Research Highlights.

Moshe Y. Vardi, EDITOR-IN-CHIEF

DOI:10.1145/1435417.1435420

True Seeds of Open Source Software

Though I appreciated Martin Campbell-Kelly's "Viewpoint" "Will the Future of Software Be Open Source?" (Oct. 2008), we must still rectify a small piece of history to apportion credit correctly. Campbell-Kelly asserted that following the release in 1964 of the IBM System/360 as a "standard computer platform" the software world experienced a radical transformation. However, the S/360 followed by a year the Burroughs B5000, a far more capable system; its successor, the B5500, came out only six months after the S/360.

The B5000 took an integrated approach to hardware and software; its architecture deliberately supported development of software exclusively in high-level languages, mainly ALGOL-based. This was at a time when the rest of the world thought a language like ALGOL could not be implemented. (Donald Knuth implemented a Burroughs ALGOL compiler on the B205 during a three-month summer vacation.) The rest of the world did not, alas, figure out ALGOL for a long time, and eventually adopted a far-less-capable language—C—as a systems programming language. In his technical report "Hints on Programming Language Design" C.A.R. Hoare said, "Here is a language so far ahead of its time that it was not only an improvement on its predecessors but also on nearly all its successors." The same can be said of the B5000.

All B5000 system software was distributed in source form, and users could read and modify it and submit patches back to Burroughs. Because it was all written in ALGOL-like high-level languages, source distribution was even more meaningful than if it was in cryptic assembler, as in the S/360. Hence, the B5000 contained the seeds of open source software.

Edsger Dijkstra's reaction is also telling; he said the week he studied the specifications of the S/360 was the "darkest week in my professional life." He later became a research fellow at Burroughs. The B5000 also had a tagged polymorphic (object-oriented) architecture designed by Robert Barton, who went on to influence Alan Kay and others in his object-oriented thinking.

The Burroughs B5000 not only predated the S/360 but reflected a far more radical transformation in software development. The S/360 was a huge backward step compared to other developments, especially the B5000, which lives on today in Unisys MCP systems.

Ian Joyner, Sydney, Australia

Campbell-Kelly responds:

Burroughs was indeed a highly innovative computer manufacturer and never achieved the sales it deserved. At the time of the S/360 launch in 1964, IBM already had over 80% of the mainframe market, compared with Burroughs' modest 4%. Despite the B5000's superb architecture and software system, the S/360 became the standard platform adopted by the great majority of computer centers worldwide.

Martin Campbell-Kelly, Coventry, U.K.

CS As Related to Philosophy As It Is to Nature

When I decided in 1984 to make teaching computer science my career, I was strongly motivated by the interconnections among CS, nature, and philosophy that have subsequently guided my approach to teaching. The "Review Article" "The Many Facets of Natural Computing" by Lila Karl and Grzegorz Rozenberg (Oct. 2008) described theoretical and experimental research that put flesh on our deepest intuition about the interdisciplinary nature of CS and thus laid a solid foundation for a computational view of nature in the CS curriculum. In this light, please consider the following anecdotal comments:

Nature as inspiration. As a young programmer in the early 1960s, I began pondering the relationship between parallel processing and the multitasking capabilities of early operating systems with our own human executor; we multitask at will, even as parallel processes run within us;

Information processing in nature. It seemed intuitive that information processing (such as genes and evolution) would take place in living things, but how about in nonliving things? For example, when two molecules combine to form a new molecule, do they exchange information? What even is information? The word's etymology suggests it is related to structure. So do two molecules have to be structurally "compatible" before they are able to combine?;

Pleasant surprise. I was a graduate student at the University of Colorado at the time professor Andrei Ehrenfeucht was collaborating with the author Rozenberg. Their theoretical research concerned how to model plant growth through formal grammars; I was in awe of their effort to try to unlock a deep mystery—a language of nature;

CS and philosophy. I invite computer scientists to consider how closely CS is related to philosophy, as well as to natural science. Philosophy contributes to CS, despite sometimes contentious disputation. Two examples are Frege's invention of first-order logic (he was, by his own account, equal parts philosopher and mathematician) and Austin's and Grice's contributions to natural language processing research. CS may likewise prove useful for philosophy. The concept of undecidability (and the use of "proof by contradiction" as a proof tool in undecidability arguments) may be useful in resolving long-standing philosophical problems; and

Philosophical intuition. The structure theorem of programming says that any (proper) program might be described using (only) control structures for sequence, selection, and iteration. My intuition tells me that when a program refers to the real world, sequence refers to time, selection to (finite) choice, and iteration to inertia.

As one who loves CS and its deep connections to nature and philosophy, I say to the authors Karl and Rozenberg, Thank You and Hooray!

Nicholas Ourusoff, New London, NH

Communications welcomes your opinion. To submit a Letter to the Editor, please limit your comments to 500 words or less and send to letters@cacm.acm.org.

ACM SIGCHI Symposium on

Engineering Interactive Computing Systems

July 14–17, 2009 | Carnegie Mellon University, Pittsburgh, USA

Conference Chair
Nick Graham, Queen's University (Canada)

Program Chairs
Gaëlle Calvary, University of Grenoble (France) & Phil Gray, University of Glasgow (UK)

Doctoral Consortium
Jean Vanderdonckt, Université catholique de Louvain (Belgium)

Local Organization
Bonnie John & Rick Kazman, Carnegie Mellon University (USA)

Publicity
Marco Winckler & Philippe Palanque, University of Toulouse (France)

Web site
Greg Phillips, Royal Military College of Canada (Canada)

Submission deadlines

All categories (except late breaking results)
25 February 2009

Late breaking results
17 April 2009

Invitation to Participate

Peoples' choices as to what products they use – from cars to games to word processors – are increasingly determined by the human-factors of the software they contain. EICS 2009 will communicate innovative engineering approaches for enhancing the quality of interactive systems. We invite your papers, experience reports and demonstrations on software processes, tools and techniques for the engineering of interactive computing systems, and look forward to seeing you in Pittsburgh in July 2009!

Topics

Original unpublished contributions are sought in areas related to some aspect of the engineering of human-computer interactive systems. Possible topics include but are not limited to:

- Interaction/interactive systems modelling
- Development processes for interactive systems engineering
- Integration of interaction design into the software development process
- Specification of interactive systems
- Requirements engineering for interactive systems
- User interface prototyping
- User interface development support (including design, implementation, testing)
- Software architectures for interactive systems
- Evaluation/testing of user interfaces
- End-user programming of interactive systems
- Dynamic generation/composition of interactive systems.

Types of Submision

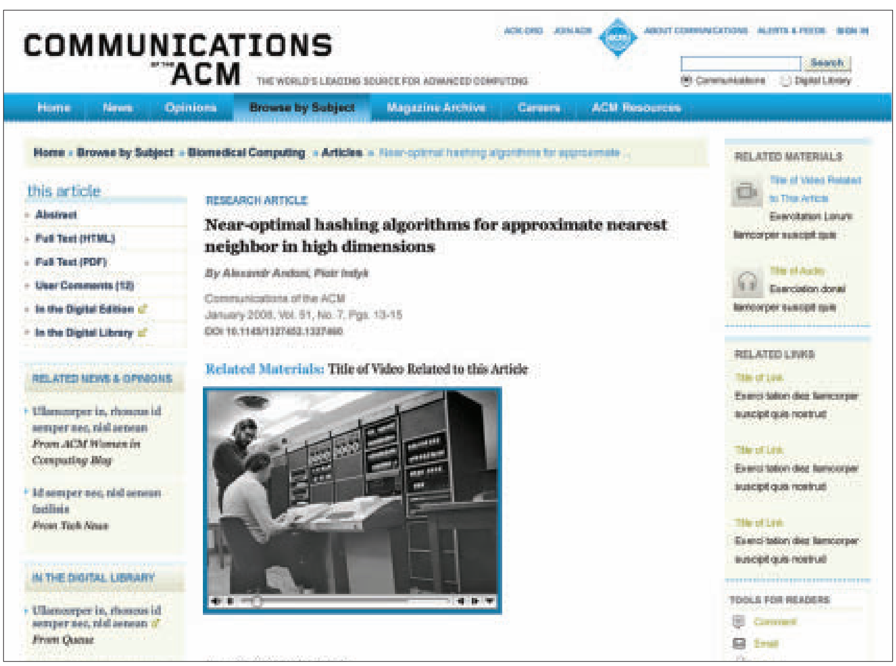
- Research Paper
- Practice and Experience Report
- Late Breaking Results
- Demonstration
- Doctoral Consortium

eics2009.org

DOI:10.1145/1435417.1435421 David Roman

Communications Preps Web-Enhanced Articles

The soon to launch *Communications* Web site will break the bounds of the printed monthly. One way it will do that is by publishing extra content with many articles from the *Communications* magazine. Every article from the magazine will be available online, and authors and contributors are enhancing the online presentation of their articles by providing supplementary material in various multimedia formats. This includes related links, sidebars, podcasts, images, videos, and other material that does not appear in the print publication. These additions take advantage of the Web's versatility, and provide readers with rich, robust content.



Webby Whatchamacallit

Although *Webster*'s dictionary doesn't define "widgets" as a Web-site feature (nor does Encarta or *American Heritage*), *Communications*' new Web site will have lots of them. Wikipedia calls widgets an "object on a computer screen the user interacts with." The widgets on cacm.acm.org will be text boxes that link readers to the site's most viewed, most emailed, most discussed articles, and to other content. Like any good design, widgets are both functional and decorative. They lead to information and enhance the look and feel of the site.

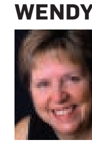
ACM Member News

SIG AWARD WINNERS

The ACM Gordon Bell Prize was awarded at the recent SC08 conference for two categories: peak performance and special achievement. A team led by Thomas Schulthess of the U.S. Department of Energy's Oak Ridge National Laboratory, won in the category of peak performance for achieving the fastest-ever performance in a scientific supercomputing application. And a team led by Lin-Wang Wang from the U.S. Department of Energy's Lawrence Berkeley National Laboratory won in the special achievement category for their research into the energy harnessing potential of nanostructures.

SIGDA presented the 2008 Pioneering Achievement Award to Edward J. McCluskey, director of the Center for Reliable Computing at Stanford University, for his outstanding contributions to the areas of CAD, test and reliable computing.

WENDY HALL SPEAKS



ACM President Wendy Hall was a keynote speaker at the 50 Years of Computing in Mexico Congress, organized by the National Autonomous University of Mexico, and held in Mexico City. Hall said the congress is a first step toward leveraging Mexico's substantial assets—its large population; exposure to international firms engaged in hardware and software marketing and manufacturing; and desire to participate in the computing revolution—to becoming a key player in computing and information technology. She said this is a transformative time for the Mexican computing community, as well as the global computing arena, and encouraged them to pursue their vision to integrate Mexico into this diverse and dynamic world.

SIGSCE 2009

The 40th ACM Technical Symposium on Computer Science Education is being held in Chattanooga, TN, from March 4–7. For more info, visit www.cs.arizona.edu/sigsce09.

news

Science | DOI:10.1145/1435417.1435422

David Lindley

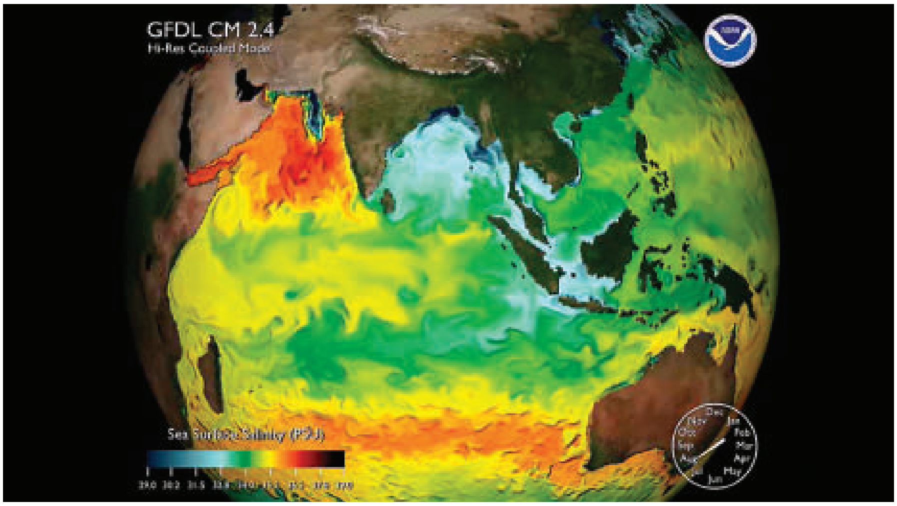
Calculating the Future

Climate researchers have no shortage of scientific issues on which to expend computer power. The biggest problem is choosing which one to tackle first.

IF YOU'RE USING a computer to solve a scientific problem, it goes without saying that additional computer power will help answer the problem faster and more accurately. Or will it? For the community of researchers who use vast computer models to simulate Earth's climate in all its glorious intricacy, greater computational capacity is always welcome, but choosing where to apply that power can be contentious. Is it better to compute existing models in finer detail, or to make the models bigger by adding more scientific content? There's no single best answer to that conundrum, and in practice the research community pursues as wide a variety of goals as it can, in the hope that a consensus will eventually emerge.

Today's so-called General Circulation Models (GCMs) include interlinked components that attempt to capture the behavior of atmosphere, oceans, sea ice, and land surface in determining Earth's climate. In computational terms, a GCM is essentially an enormous and intricately interlinked collection of ordinary and partial differential equations that calculate air and ocean currents and their associated heat flows; the absorption of the sun's heat (which depends on cloud cover and the amount of snow and ice covering the planet's surface, among other things); the radiation of heat from land and sea ice back into the atmosphere; humidity and precipitation; and a great deal more. Typically, these models cover the planet's surface by calculating at grid points spaced approximately 100 kilometers apart, and divide the atmosphere, up to a height of some 15 kilometers, into perhaps 20 layers. From a global perspective, with Earth's total surface area amounting to just more than a half-billion square kilometers, that's a lot of grid points, but it takes no scientific expertise to understand that weather conditions can vary significantly across hundred-kilometer distances. As a result, many medium-scale phenomena in current GCMs cannot be calculated directly but must be dealt with by "parameterization," meaning that important aspects of small-scale physics are in essence approximated and averaged over grid cells.

An obvious use of greater computer power is to decrease the distance between grid points. That's particularly valuable in ocean modeling, says Ben Kirtman of the Rosenstiel School of Marine and Atmospheric Science at the University of Miami, because calculating on a grid spacing of a few kilometers would directly capture important heat and current flows, without parameterization. Kirtman is working with a project recently funded by the National



A U.S. National Oceanic and Atmospheric Administration climate change model that couples a 25 km-resolution ocean model with a 100 km-resolution atmosphere model.

IMAGE COURTESY TONY ROSATI AND TOM DELWORTH, NOAA/GFDL

A climate change model of Earth with a quasi-uniform but nonorthogonal quadrilateral grid.

IMAGE COURTESY OF M.A. TAYLOR, J. EDWARDS, A. ST.CYR, "PETASCALE ATMOSPHERIC MODELS FOR THE COMMUNITY CLIMATE SYSTEM MODEL: NEW DEVELOPMENTS AND EVALUATION OF SCALABLE DYNAMICAL CORES", J. PHYS. CONF. SER. 125 (2008) 012023

Science Foundation to apply petascale computing capacity—10^{15} floating point operations per second—to the analysis of ocean-atmosphere interactions. He cites the example of tropical instability waves in the eastern Pacific Ocean as a medium-scale marine phenomenon that climate scientists "originally thought the atmosphere didn't care about." Higher-resolution calculations show, however, that these instability waves, along with mesoscale ocean eddies measuring 10 kilometers or so across, profoundly influence not only how heat mixes both horizontally and vertically within the ocean, but also how heat is exchanged between ocean and atmosphere. The eastern Pacific Ocean-atmosphere interaction, Kirtman explains, in turn feeds into the year-to-year evolution of the well-known El Nino-Southern Oscillation, demonstrating that regional calculations on the kilometer scale are crucial to a better understanding of globally significant phenomena.

Atmospheric Challenges

The atmosphere presents more difficult problems. Different GCMs are often compared in terms of the average global temperature increase they predict for a doubling of atmospheric carbon dioxide. That figure ranges from approximately 1.5° C to 4.5° C, and much of the variation between models stems from the different ways they parameterize fine-scaled atmospheric features such as convection and cloud cover. Higher-resolution calculations will do much to clarify convective and turbulent flows, says Jerry Meehl of the Climate and Global Dynamics Division at the National Center for Atmospheric Research (NCAR) in Boulder, CO, but clouds are more complicated. Clouds reflect sunlight from above but trap heat rising from below, so their net effect on climate depends on details of cloud composition and structure that current models struggle to depict. Typically, models allocate some percentage of various cloud types to each grid cell, and allow some randomized overlap of cloud layers at different altitudes. But the biggest obstacle to more accurate modeling, says Meehl, has been a lack of detailed observations of the way clouds literally stack up in the atmosphere. In this case, increased computer power will only be useful if it is coupled to better physical data on cloud structure and properties that can be used to refine cloud simulations. "The cloud community now is as excited as I've ever seen them," Meehl says, because satellites are beginning to provide just the type of detailed 3D data that modelers need.

The steadily increasing resolution of GCMs is blurring the already fuzzy distinction between weather and climate. Researchers are beginning to calculate models with 50-kilometer resolution over periods of decades, enabling them to see how climate change might affect the frequency and intensity of extreme storms or the statistics of droughts. Such information, rather than the more abstract concept of global average temperature, starkly conveys the tangible consequences of global warming.

In addition to using computing power to calculate on an ever-finer scale, climate researchers can always think of more science to put into their simulations. Historically, the growth of computational capacity allowed researchers to integrate previously separate models of ocean, atmosphere, sea ice, and land, and that trend continues on a number of fronts. At the moment, for example, atmospheric carbon dioxide concentration is applied to climate models as an external parameter, derived from the work of scientists who add up emissions from tailpipes and smokestacks and, taking into account the natural processes that absorb and release the gas, try to estimate how much CO_2 will be in the atmosphere 10, 20, or more years from now. But this approach misses all types of crucial feedbacks. Changing temperatures of the oceans affects how well they hold dissolved CO_2, while changes in the world's vegetation cover, due to a warming climate, influence the amount of carbon that ends up in the atmosphere rather than being taken up by biomass. Climate modelers are beginning to integrate parts of this complex network of feedbacks into GCMs, so that ultimately they will be able to input human CO_2 emissions directly into the models, and allow the computer to figure out where it all ends up—and how that disposition changes in a changing climate.

Climate researchers, then, are forced

Education

Students Flock to Game Design

As governments and universities worldwide develop strategies to reverse a dizzying downturn in computer science students, a hot field of study is getting even hotter and helping to rekindle interest in computer science.

Game design has become a popular new major at more than 200 schools across the U.S., according to a report from the Entertainment Software Association. Because game creation crosses several disciplines, the diversity of programs that offer such courses is stunning: Fine arts colleges, engineering schools, film schools, music schools, and even drama programs are sending graduates into the burgeoning industry.

Game companies literally recruit "armies" to work in studios around the world, according to the *Los Angeles Times*. The jobs vary from inventing characters, to writing dialogue, composing music, creating digital scenes, and writing the software that rules these fantasy worlds. A blockbuster game can take more than 100 developers, each working for two or more years to complete one product.

Colleges began noticing the game industry about six years ago when its sales started to compete with movie box-office receipts, says Jessie Schell, who teaches game design at Carnegie Mellon University. Since then, computer science schools have experienced a boom in the number of programs to train future developers.

to make compromises when deciding what to do with more computing power. Modelers have consistently aimed to get five or 10 calculated climate years per day of computing time, says Meehl, and keeping to that figure sets a practical limit on the increase in resolution or scientific complexity that additional computing power will buy. To get climate change predictions right, Meehl says, new science must eventually be included, because inadequate treatment of various feedbacks is "in large part what contributes to the disagreements among models." On the other hand, Kirtman says that he prefers to focus on refining what's already in the models, because "we can't even get the clouds right and until we do that we can't usefully add in the other feedbacks."

Is it better to compute existing models in finer detail, or to make the models bigger by adding more scientific content?

Programming Problems

Getting everything right is still years away. Achieving global one-kilometer resolution with current GCMs—while adding no new science—is a computational task of exascale proportions, requiring the performance of approximately 10^{18} floating point operations per second. Right now, climate modelers are beginning to grapple with petascale systems, built from tens of thousands of processors. But making good use of such a system is no easy matter as the evolution of efficient programming techniques has not kept pace with the growth of computing power, says Dave Bader of the Program for Climate Model Diagnosis and Intercomparison at Lawrence Livermore National Laboratory.

What makes coding a GCM a particular challenge, Bader explains, is the huge amount of information that must be continually and rapidly transferred among the model's numerous components. On top of that, climate models generate large amounts of output data, and getting those results out of the program and into displays that users can make sense of is also challenging.

If the limiting factor in running a climate model on a multiprocessor system is inefficient communication of information within the program, then the amount of processing power dedicated to solving equations falls and the model fails to take advantage of the raw processing power available. "The programming model we use [now] is not viable anymore in the next couple of generations of computers," says Bader. The handful of vendors in the supercomputer market—IBM, Cray, and a few others—don't devote as much effort as they used to in developing languages and compilers that serve scientific users, he adds, so that responsibility for such things falls increasingly on the shoulders of researchers at various laboratories working in partnership with the vendors.

The drive for programming efficiency has changed the way climate scientists work. Meehl says that when he started in the 1970s, scientists would "fiddle with Fortran code, submit it to the machine, and it just ran. We didn't have to think about it a whole lot." Now, though, the team behind NCAR's Community Climate System Model (CCSM), a state-of-the-art GCM used by researchers across the U.S., includes a working group of software engineers dedicated to ensuring the code runs reliably and efficiently.

Still, there's room for innovation. "I actually do a fair amount of my own code work, by the seat of my pants," says Kirtman. His programming may be inefficient and prone to failure, but that's not important while he's developing it, Kirtman says. To get his new work on ocean-atmosphere interactions incorporated into the CCSM, he turns it over to software engineers who transform his pilot program into a robust piece of modeling that any researcher can download and use. That's something he couldn't do, Kirtman says, and the net result is "I get a lot of feedback from people who are trying to apply my methods to their problem—that's really powerful to me." C

David Lindley is a science writer and author based in Alexandria, VA.

Network Architecture

Inside Internet Hardware

The idea was simple, only no one thought it was attainable. Surely network hardware companies would never allow outside researchers to access their equipment and established installation, right?

Not so, according to *Technology Review*, reporting on a project called OpenFlow that has opened some of the most commonly used network hardware from companies such as Cisco, Hewlett-Packard, Juniper, and NEC. A team of researchers, led by Stanford University's Nick McKeown, has been given unprecedented access to Internet hardware and an opportunity to improve the Internet's security, reliability, energy efficiency, and pervasiveness.

"In the last 10 years, there's been no transfer of ideas into the [Internet] infrastructure," said McKeown, a professor of electrical engineering and computer science. "What we're trying to do is enable thousands of graduate students to demonstrate ideas at scale. That could lead to a faster rate of innovation, and ultimately these ideas can be incorporated into products."

The Stanford team secured permission from major equipment vendors to write a small amount of code that grants access to the flow table. When a packet of data arrives at a switch, for instance, software in the switch looks up instructions on the flow table to decide where to send the packet. OpenFlow essentially offers direct access to the flow table to add and delete instructions. As simple as it sounds, McKeown explains it has not been implemented before because the prevailing assumption was that vendors would not open their hardware.

The Stanford team sees the potential to open up the airwaves, allowing portable devices to access any wireless network they can detect. The goal, McKeown says, is "seamless mobility. We'd love to come up with a way to rearchitect cellular networks. But that's further out. We're talking 10 years."

Society | DOI:10.1145/1435417.1435424

Jeff Kanipe

The Universe in Your Computer

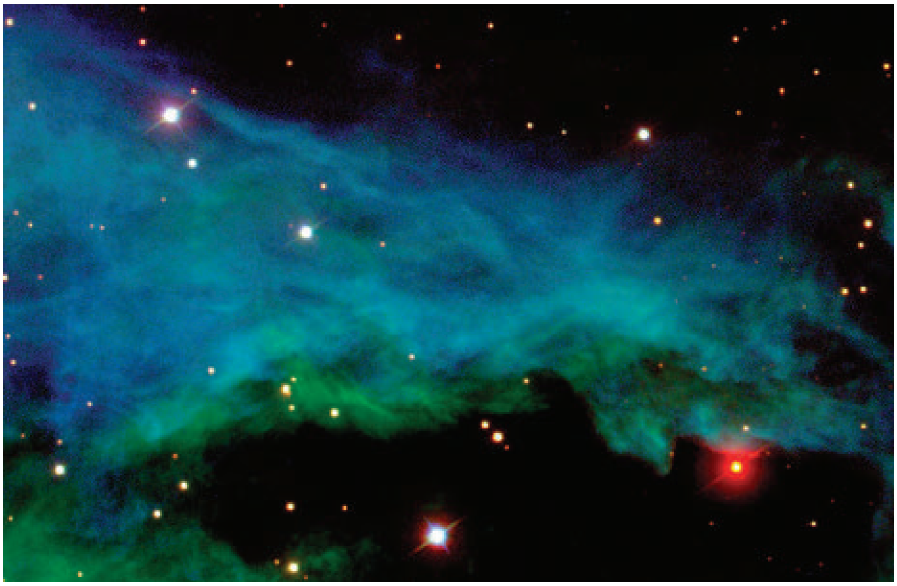
Two virtual astronomical telescopes promise to transform the way people view and study the cosmos.

THE TELESCOPE, IT is often said, is like a time machine in that when you focus on a star, you see that star as it actually appeared tens or even thousands of years in the past. That's how long it takes for light—zipping along at a speed of 300,000 kilometers per second—to cross the vast gulf of space between "there" and "here." These look-back times, as astronomers sometimes call them, can reach astonishing proportions. When you focus your backyard telescope on something like the Andromeda Galaxy, for example, you are looking at light that has been traveling more than two million years to reach your eye. But that's nothing compared with the purview of the world's most powerful telescopes, such as the W.M. Keck, located near the summit of Mauna Kea on the Big Island of Hawaii, or the Hubble Space Telescope, orbiting above the distorting effects of Earth's atmosphere. These time machines have revealed galaxies that appear as they were more than 13 billion years ago—or 13 billion light-years away—an epoch in time when the universe was only about 800 million years old.

It boggles the mind, such space and time. There are simply no adequate words to describe the enormity of the universe or its spectacular, never-ending assortment of stars, nebulae, and galaxies. But now, two powerful Web applications—Microsoft's WorldWide Telescope (WWT) and Google Sky—are providing users with new ways to explore the universe. Both essentially turn a personal computer into a multiuse virtual telescope.

Zooming In

Last May, Microsoft launched the beta version of its WorldWide Telescope, making it the most recent desktop telescope available to the public. At the time, Microsoft chairman Bill Gates called the WWT "a powerful tool for science and education that makes it possible for everyone to explore the universe." The application itself is not a simple browser with links, but an integrated amalgam of data and images from such surveys as the Digitized Sky Survey, the Sloan Digital Sky Survey, the Hubble Space Telescope, the Chandra X-ray Observatory, and the Spitzer Space Telescope. Thus, images are available across multiple wavelengths of the electromagnetic spectrum. The application, which is coded in Microsoft's C# .NET and created with Microsoft's Visual Experience Engine, is a combination of software and Web services that allows users to pan smoothly across the sky while accessing terabytes of images and data from multiple sources. Microsoft likens the result to a "media-rich, immersive experience," with applications for both amateur and professional astronomers.



Microsoft's WorldWide Telescope contains a wealth of data and images, such as this partial view of the Orion Nebula from Sloan Digital Sky Survey.

Microsoft's WorldWide Telescope enables users to pan smoothly across the sky and access terabytes of images and data from multiple sources.

Users can zoom in on the Orion Nebula, for example, and cross-fade from one wavelength view of the nebula to another, revealing hot pockets of gas, which unaided human eyes cannot visually detect, and young stars embedded in obscuring clouds of dust. Crossfading provides a powerful method of literally looking into the environments and dynamics of celestial objects, thus providing key insights into their nature and why they look the way they do. Menu tabs allow serious researchers to access professional database archives of studies on the objects of their choice. Users can also find the current locations of planets, observe stunning panoramas taken by rovers from the surface of Mars, and zoom in on rotational views of the Moon, Venus, Earth, and Jupiter and its moons Io, Ganymede, and Callisto.

The basic layer of the northern sky in both the WWT and Google Sky is comprised of sky surveys conducted over the years at Palomar Observatory in California, while the southern sky is derived from surveys at the Anglo-Australian Observatory in Australia. A digital version of the photographic plates of the Palomar survey, called the Digital Sky Survey, was produced as a collaborative project between the Space Telescope Science Institute in Baltimore, MD, and the California Institute of Technology. The project's principal investigator, S. George Djorgovski, is excited that the general public can now easily view the science data set that is available in both programs. "I think Google Sky and WWT are great public outreach venues. I really feel positive about them," says Djorgovski. "It is probably too early to gauge any social impact of these [programs], and it will be tricky to measure in any case. But it's got to be good, for obvious reasons."

The WWT has its origins in the pioneering work of Jim Gray, whose many computer science-related accomplishments include developing the SkyServer program for the Sloan Digital Sky Survey, which is considered the most ambitious astronomical survey of the sky ever undertaken. (For more about Gray's astronomical contributions, see the November 2008 *Communications* article, "Jim Gray, Astronomer," by Alexander S. Szalay.) SkyServer is the forerunner of an information software architecture known as contextual narrative, which seeks to integrate data and stories with interactive contextual exploration in a one-stop shopping environment.

As an integral part of the WWT, contextual narrative both enhances and facilitates the learning experience, giving it new dimensions, says Curtis Wong, principal researcher of the Microsoft Next Media Research group. "It's enabling a new generation of kids who learn through interaction to be inspired to explore the universe in context, guided by astronomers and seamless links to the world's information," Wong says. "I think learning in context has been the original goal for WWT since the beginning, and that has the biggest potential impact for the users."

Google Sky offers multiple information layers, including constellation figures and the current positions of planets.

Information Rich

In August 2007, Google launched its universal Sky browser by making it available as a free download at Google Earth. Like the WWT, it also provides a seamless view of the night sky using images from the Digital Sky Survey, the Hubble Space Telescope, and other professional observatories. Google Sky also uses multiple information layers that can be selected under its sky database, including constellation figures, the current positions of the planets, and a backyard astronomy layer that labels stars, constellations, and celestial objects. When the "education center" and "current sky events" options are selected, users can click for Hubblecast videos, virtual tours of stars and galaxies, and detailed descriptions of celestial objects provided by other sources, such as NASA's Hubble site and Wikipedia. When all layers are selected, though, the screen can become crowded, but the numerous choices make Google Sky undeniably information rich.

Google Earth was created to project imagery onto the surface of a sphere. For Google Sky, that perspective is reversed by using the same infrastructure to project images of the sky onto the inside of a sphere, creating a re-

Telecommunications

Payment Via Wallet Phone

Wallet phones may still be viewed in most nations as a futuristic hopeful, but in Japan more than 50 million cellphone users are already carrying mobile money models. Reuters reports Japan has pioneered not just the technology for wallet phones, but also the business models that will pave the way for the devices to become a standard payment method in the future. Some 700 million people worldwide are expected to own wallet phones by 2013.

Today more than half of all cellphone users in Japan are carrying phones capable of serving as wallets. Success in Japan and in trials abroad have shown that the technology is ready for cellphones to replace credit cards, cash, as well as serve as transportation and movie tickets and electronic keys for homes and offices. Indeed, the world's biggest payment card company, MasterCard, announced last fall it was in negotiations about commercial launches of wallet phones with several banks, and during the next two years it expects to see substantial activity from retail-focused banks.

But there are still hurdles to overcome before success is a given, particularly in terms of breaking the psychological barrier for consumers skeptical about using cellphones as credit cards. There is also the major obstacle of working out new business models as the lines blur between banks, financial institutions, and cellphone companies.

Despite Japan's relative success with wallet phones, still only one-third of wallet-phone holders use their cells for purchase. Consumers in their 20s and 30s are the main users of wallet-phone services. Research shows that once they start using, they tend to use frequently and repeatedly, making it a useful tool for companies to track their customers and shopping habits.

McDonald's Japan and 7-Eleven are testing mobile discount coupons, and a joint Sony-DoCoMo venture has launched a mobile platform for retailers to offer such services.

alistic representation of the celestial vault. There is, however, a small tradeoff (which some users argue is not so small). Because Google Sky uses a latitude and longitude projection, the stars in the original images were significantly distorted between seven and eight degrees of both celestial poles. Hence, these regions were replaced with a lower-resolution view of the sky derived from the Tycho II star catalog. The stars in those polar regions are obviously not as sharp as the other parts of Google Sky's sky (they exhibit a decided radial stretch from the pole outward), but they are properly scaled and their colors are based on real color data.

Like Google Earth, Google Sky utilizes Keyhole Markup Language (KML), which is an XML-based language for displaying geographic data and visualizations for Web-based 3D browsers. In this case, KML files display not mountains and cities, but celestial objects as well as annotated data files. Users can add their own content by converting it into a KML file and posting it at either Google Sky or on the Web so that others can add it, if they choose. One such example is an orrery that shows the positions of the planets in the solar system with respect to each other on a particular night.

"I'm coming across this problem that the kids are way ahead of teachers," says Carol Christian.

Lior Ron, a product manager for Google Sky, says the program encourages users to see Earth in perspective with the vastness of the universe. "We hope that Google Sky will bring about a fundamental change in the way we perceive our place in the universe, just as Google Earth changed how we look at our planet," he says. "A number of [theorists] have talked to us about how looking at things like the IRAS [infrared astronomical satellite] infrared map in the Web version of Google Sky really brought a physical reality to their mental model of the sky that they hadn't had previously," says Ron.

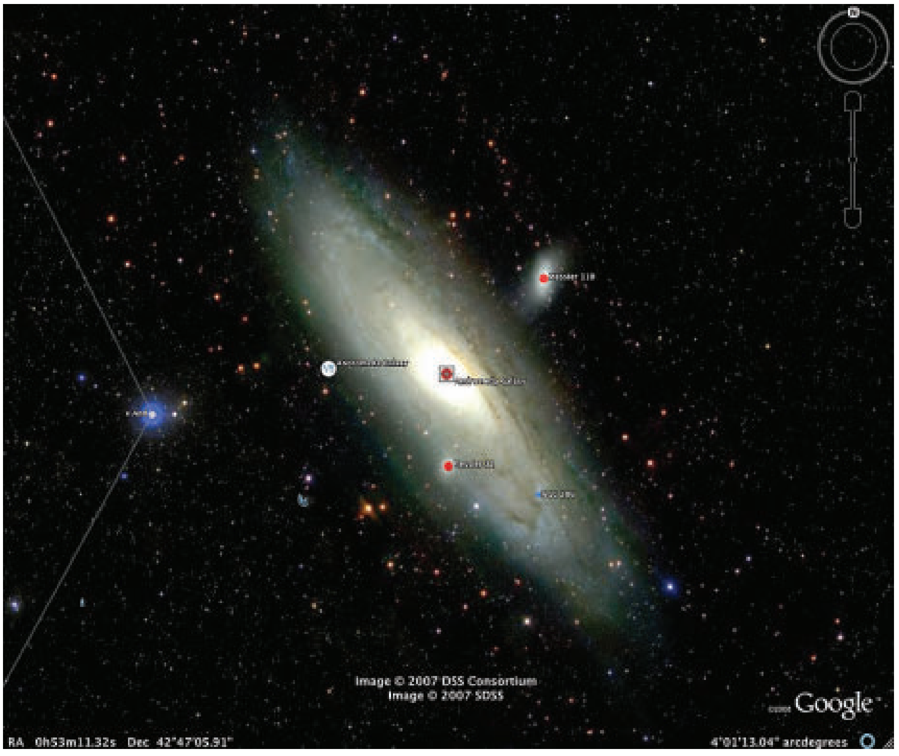
One of the most fascinating aspects of WWT and Google Sky is how they allow you to view celestial objects in context with their surroundings in the immensity of space. For example, when you zoom in on the region of sky known as Hubble Ultra Deep Field, you'll discover it is just one-tenth the diameter of the full Moon, small enough to cover with a pencil tip held at arm's length. But when you follow the clickable links, you'll discover this tiny bit of celestial real estate contains more than 10,000 galaxies.

Field of Dreams

The advent of these virtual telescope programs is being celebrated by astronomers, but as Djorgovski notes, measuring their impact on education is tricky. Carol Christian, an astronomer at the Space Telescope Science Institute, home of the Hubble Space Telescope, agrees. "It's something that's going to take years," she says. "This isn't something you can do in a couple of months. Right now it's a field of dreams. We built it. Will they come?"

Christian believes the key to Google Sky's success will be innovative educators. "I'm coming across this problem that the kids are way ahead of teachers," she says. "We need to get teachers. . . to understand that the student of the future and the worker of the future needs to be facile with finding information, analyzing it, applying critical thinking, making decisions, and finding the data they need to answer a question."

To date, both the WWT and Google Sky claim millions of active users. If these numbers are any indication, both applications will soon be an essential part of the science classroom and museum, as well as a powerful tool that will enable researchers to access, publish, and update data in context with the very universe they study. The virtual telescope is a field of dreams with unlimited possibilities, all of which will be explored and augmented by users in ways we can only speculate about now. In the not-too-distant future, computer scientists may look back on the rise of applications such as the WWT and Google Sky as forerunners of the much ballyhooed Web 3.0 era, which would be fitting since it promises to be light-years ahead of today's Internet.



A view of the Andromeda Galaxy, a spiral galaxy located more than two million light-years away, as it appears in Google Sky.

Based in Derwood, MD, **Jeff Kanipe** is the author of *The Cosmic Connection: How Astronomical Events Impact Life on Earth.*

Technology | DOI:10.1145/1435417.1435423

Alex Wright

Get Smart

Future generations of smartphones will be context aware, tracking your behavior, providing information about the immediate environment, and anticipating your intentions.

The problem with today's smartphones is that there's nothing terribly smart about them. Even the most souped-up devices amount to little more than stripped-down PCs, with tiny screens and maddening keyboards. The era of the PC-in-your-pocket may soon give way to a new type of smartphone, however, that is less of a portable workstation and more of a personal digital assistant that anticipates what you need to know, and when.

The next generation of context-aware smartphones will take advantage of the growing availability of built-in physical sensors and better data exchange capabilities to support new applications that not only keep track of your personal data, but can also track your behavior and—this is where the truly smart part will finally come into play—anticipate your intentions.

Many smartphones already contain the basic building blocks for context awareness such as physical sensors, like GPS, accelerometers, and light sensors, coupled with operating systems that allow developers to create their own applications. What's missing, however, has been the middleware that will enable applications to juxtapose information about your physical location with data from other applications. "A lot of the basic technology has existed for years," says Erick Tseng, product manager for Google's Android. "The challenge has been that developers didn't really have access to those devices."

The first wave of context-aware applications will focus on pulling in data keyed to the user's physical location. "Your phone is going to be a sensor," says Ken Dulaney, a distinguished analyst at Gartner Group. Equipped with magnetic compasses, accelerometers, and GPS, smartphones will begin to support augmented reality applications that draw on detailed information about the location and orientation of your phone.

Enkin, a 3D navigation system built for Google's Andriod, provides a live mode via a video feed (left), a landscape mode (top right), and a map mode.

IMAGE COURTESY OF MAX BRAUN & RAFAEL SPRING

The result will be a type of portable data X-ray device that reveals information about any location at which you point your phone.

San Francisco-based GeoVector is one of several new companies developing software to take advantage of more accurate location-sensing data. "We want to convert the phone into a virtual mouse that you can use to click on the real world," says Pam Kerwin, GeoVector's head of strategic business development. Using a combination of latitude and longitude data, provided by GPS, and information about the user's orientation, which is gleaned from an electronic compass, the software creates a virtual vector that can superimpose location-specific data on a mobile phone screen.

Similarly, Enkin—a 3D navigation application based on Google's Android—takes advantage of location data to allow users to scan the physical environment using GPS and compass data. In live mode, it works with a phone's built-in camera to project onscreen annotations of physical locations atop a live video feed, by querying a database of location data tagged with latitude and longitude coordinates. So, if you point the device at a hospital building, the term "Hospital" appears superimposed over the video image in real time.

Future smartphones will provide information via a live video feed about locations at which the phone is pointed.

While advanced location awareness will open up new avenues for application development, this kind of physical sensing constitutes the most simplistic level of context awareness. As smartphone platforms mature, they will start to merge physical location data with information about user behavior—drawn from calendars, email, text messages, and other applications—to begin modeling your intentions. "The trend is toward long chains of integration," says Dulaney.

For example, a context-aware phone might know that you're sitting inside a movie theater and automatically mute itself. And if your calendar knows that you're about to leave for a meeting on the other side of town, your phone could query a public traffic database and determine that you're going to be gridlocked, then notify colleagues that you'll be delayed. Or if you're traveling and it's dinnertime, your smartphone might suggest a nearby restaurant based on your location and previous dining history.

Researchers at PARC, based in Palo Alto, CA, have proposed four levels of system awareness: basic context awareness (including location, time, and other details of the physical environment); behavior awareness (typing, walking, standing, or clicking a button); activity awareness (shopping, dining, or traveling); and intent awareness (predicting the future). "Intention modeling is the Holy Grail for context awareness," says Bo Begole, a principal scientist at PARC who manages ubiquitous computing initiatives. Begole's team has been working on a prototype mobile application called Magitti (a contraction of "magic lens" and "graffiti") that monitors users' behaviors across a wide swath of applications on both desktop and mobile devices, including calendars, email, text messages, and Web browsing history, to give users recommendations of where to go and what to do in any particular location.

In a similar vein, Intel researchers have developed a prototype of a smartphone-based system that draws data from multiple applications and matches it against location data. One demonstration shows a user walking into a conference room, automatically triggering the location-aware smartphone to check the user's calendar, determine whether the user is in charge of the meeting and, if so, request that the conference room computer begin displaying his or her stored presentation on the room's projector.

With intention modeling, users will receive recommendations based on both their computer and mobile device activity and on their current location.

Major Hurdles

For now, most of these context-aware mobile applications remain in the R&D stage. To bring contextual computing to the masses, phone manufacturers and software vendors will need to overcome some major hurdles. Not the least of these is the lack of open standards for exchanging context data between applications. In the absence of such standards, several companies are pursuing their own implementations, which demonstrate how this kind of contextual integration might eventually work.

Google's Android relies on a middleware-like system of "intents" and "listeners" tied to each application. For example, an incoming phone call could send out an intent that could be picked up by any number of listening applications—like the phone receiver, email, or calendar—which could then notify other applications of the person's activity. Users can establish the priority of listeners by setting preferences in individual applications. Developers can also create their own intents and listeners within any application.

At Intel, researchers have proposed a Context Aware Computing framework that allows for platform-level integration of contextual data that relies on an aggregator component managing context information from multiple sources. An access control policy enforcer manages permissions, while an analyzer executes rules, allowing clients to establish algorithms for interpreting context information. At the hardware level, a simple dynamic link library enables the phone to access sensors transmitting information about data such as location, orientation, and identity. At the software level, a set of Web services allows the phone to interface with applications anywhere in the cloud of accessible applications. Finally, a level of middleware glues the whole thing together, joining data from the sensors and applications with user input, storing contextual information, and allowing the phone to share that data across applications or even between different devices. The framework would operate more effectively, however, if it were integrated into an operating system. "We think these services could be driven further down into the system," says Lester Memmott, a software architect in Intel's Software Pathfinding and Innovation Division.

These prototype implementations allow developers to explore the possibilities of integrating contextual data, but the era of cross-platform interoperability seems a long way off. In today's fractured smartphone market, a slew of competing operating systems battle for dominance, including those of Apple, Microsoft, Research in Motion, Symbian, and a variety of Linux variants, including Google's Android, Intel's Moblin, and LiMo. Beyond the operating systems lies a babel of competing standards including Bluetooth, the Digital Living Network Alliance, Universal Plug and Play, and ZigBee, all requiring developers to create custom data interfaces for each type of device.

Right now, there seems to be little incentive for cooperation about developing context-aware applications. However, one ray of hope may come from the Open Handset Alliance, whose members—including Google, Intel, LG, Motorola, Samsung, Sprint, T-Mobile, and other major players in the mobile phone market—are taking tentative steps toward exploring a common framework for sharing contextual data.

Beyond the technical and business hurdles, perhaps the greatest challenge will come from the most unpredictable variable of all: users. In a world where smartphones start to collect and share more personal data about users' behaviors and preferences, people may begin to feel increasingly uncomfortable with the idea of a smart device that seems to be tracking their daily activities. Regardless of whether such concerns are well founded, they may prove difficult to overcome.

"We need to address these concerns in a responsible way," says Memmott. How can developers help to ameliorate users' privacy concerns? "Transparency is the key," says Begole, who advocates giving users control over their data and especially providing users with tools to manage the sharing and presentation of personal data.

If developers and manufacturers can work together to overcome these technical, business, and behavioral obstacles, the next generation of mobile devices may finally start living up to their brainy potential. The term "smartphone" may still be a misnomer, however: Smart they will undoubtedly be, but will anyone still care about the phone?

Alex Wright is a writer and information architect who lives and works in New York City.

News | DOI:10.1145/1435417.1435441

Michael Ross

A Pioneer Woman

Programmer Jean Bartik is inducted into the Computer History Museum's Hall of Fellows.

JEAN JENNINGS BARTIK came from a long line of teachers and her most supportive college professor encouraged her to become a teacher, but when she graduated as the only math major from her war-depleted college in 1945, she had no interest in a classroom career. Raised on a farm two miles outside tiny Alanthus Grove, MO, she yearned for travel and excitement. Bartik responded to a government ad for women math majors and only a few hours after being accepted, she was on an eastbound train to faraway Philadelphia, PA.

The government job turned out to be for a human "computer" and involved laboriously calculating artillery projectile trajectories that soldiers used to aim new Army guns. Bartik loved city life, but the work itself was tedious and repetitive. So, she jumped at the opportunity to move, sight unseen, to a secret government project. Thus, the Missouri farm girl found her dream job as one of the world's first computer programmers, joining a team of six women who were charged with figuring how to program the world's first general-purpose, electronic, and digital computer, the Electronic Numerical Integrator And Computer (ENIAC), which was designed to calculate artillery trajectories 1,000 times faster than a human computer.

In recognition of her pioneering work, Bartik recently received a 2008 Fellows Award from the Computer History Museum and spent an evening at the Mountain View, CA, museum regaling an audience with stories about her professional life.

There were no manuals or instruction guides for programming the ENIAC, which included 17,468 vacuum tubes, occupied more than 680 square feet, and weighed 30 tons. Instead, Bartik and her five female colleagues—Kay McNulty Mauchly Antonelli, Betty Snyder Holberton, Marlyn Wescoff Meltzer, Frances Bilas Spence, and Ruth Lichterman Teitelbaum—pored over its logical and electrical block diagrams and discussed design details with the male engineers and physicists who had created them. Ultimately, the women figured out how to set ENIAC's 3,000 switches and hundreds of connection cables so calculations would progress correctly through the complex machine.

PHOTOGRAPH COURTESY OF THE COMPUTER HISTORY MUSEUM

Bartik claimed that she and her colleagues excelled at their jobs because, unlike many of the visionaries involved in the ENIAC project, the women were good "finishers"; they would not stop until all the loose ends were nailed down.

Although the women were classified by the Army as "sub-professional," working for ENIAC's inventors, John Mauchly and J. Presper Eckert of the University of Pennsylvania, was like being in a "technical Camelot," Bartik said. While as intense as any present-day startup, the work environment was inspiring and respectful of everyone's contribution, regardless of gender or race. Unfortunately, this was not true elsewhere, and the women's contributions were not recognized for decades.

Only three of the women programmers—Antonelli, Bartik, and Holberton—continued to work in the computing industry after ENIAC. In 1947, Bartik created and led the team that converted ENIAC into a stored program computer and, at the Eckert & Mauchly Corp., played a key role in several aspects of the UNIVAC 1, the first commercial computer. (Holberton died in 2001; Antonelli died in 2006.)

Bartik said the working conditions turned into "a job from hell" after Remington Rand bought the financially troubled Eckert & Mauchly Corp. in 1950, and she decided to concentrate on family life and spent the next 16 years raising her three children.

When Bartik returned to the computer industry in the mid-1960s, minicomputers were very popular, and she worked for a number of companies as an analyst, editor, and in customer support. Never one to shade the truth as she saw it, Bartik sometimes found herself at odds with managers and marketers, but every so often she'd smile when she recognized a "new" technology as something she'd discussed years earlier with the ENIAC pioneers.

The contributions of Bartik and the women ENIAC programmers might never be known today had not a Harvard student, Kathryn Kleiman, noticed a passing mention of them in a computer history autobiography in 1986 while researching a paper on women in computing. Now a lawyer, Kleiman has produced 20 hours of broadcast-quality, oral-history interviews and, with a grant from ACM's SIG Discretionary Fund, created an eight-minute trailer for a full-length documentary on the ENIAC programmers.

After reflecting on the ups and downs of her career, Bartik offered a positive, personal message for those entering today's computer industry: "There will always be people thinking outside the box—thank goodness! Enjoy every moment you have. You're as happy as you choose to be. So I choose to be happy."

Michael Ross writes about science and technology from San Jose, CA.

DOI:10.1145/1435417.1435426 Jonathan Zittrain

Law and Technology
The End of the Generative Internet

Exploring the expectations and implications for version 2.0 of the Net's new gated communities.

I recently wrote a book about the future of the Internet. The book's thesis is that the mainstream computing environment we've experienced for the past 30-plus years—dating from the introduction of the first mainstream personal computer, the Apple II, in 1977—is an anomaly. The basic building blocks of modern IT are PCs that anyone can reprogram, connected to an Internet that unquestioningly routes bits between two arbitrary points. This has led to a generative revolution where novel and disruptive technologies have come from obscure backwaters—and conquered. While incumbents bet on (or were) gated-community networks like CompuServe, Prodigy, and AOL, or makers of "smart appliances" such as dedicated word processors and video-game consoles, dark-horse candidates like the Internet and the PC emerged unexpectedly and triumphed, helped along by commercial forces that belatedly hopped on their bandwagons.

So why anomalous? Technologies that allow—indeed, depend on—contributions from anywhere and trust of unknown contributors at first thrive in elite communities where people have the technical astuteness to know how to manage them and the ethos of tinkering and mutual assistance. As the platforms' popularity increases and they enter the mainstream, greater numbers and a decline in the average user's skill set make them increasingly worth subverting. For example, spam is more profitable when there are hundreds of millions of recipients instead of thousands, and worms and Trojans have that much more personal data to mine or processors to hijack when over one billion PCs are in use.[1]

Imagine if Microsoft had adjusted Windows to act the way the iPhone and Facebook apps platforms do.

One approach to the problem is to try to double-click our way out of it: to subscribe to antivirus software that tries to stop harmful code from running on our machines. Such Patriot missile-style defense aspires to keep bad code as a mere nuisance, but it depends on 100% accuracy and a willingness by users to defer to the recommendations of their antivirus packages—difficult when there are plenty of false positives.

Another approach is to eliminate or qualify the generative character of our technology. Apple's iPhone was introduced in 2007, and its first version, like its near-twin iPod, allowed no outside code at all. As the book went to press in October 2007, Apple, Inc., provided perfect bookends for the trajectory of the consumer information technology ecosystem: "Rather than a platform that invites innovation, the iPhone comes preprogrammed. You are not allowed to add programs to the all-in-one device that Steve Jobs sells you.

ILLUSTRATION BY JOHN HERSEY

Its functionality is locked in, though Apple can change it through remote updates. Indeed, to those who managed to tinker with the code to enable the iPhone to support more or different applications, Apple threatened (and then delivered on the threat) to transform the iPhone into an iBrick. The machine was not to be generative beyond the innovations that Apple (and its exclusive carrier, AT&T) wanted. Whereas the world would innovate for the Apple II, only Apple would innovate for the iPhone. (A promised software development kit may allow others to program the iPhone with Apple's permission.)"[3]

The parenthetical remark has since come true (the iPhone Software Development Kit was released in March 2008), and as of this writing in late 2008, several months after the introduction of iPhone 2.0, it is difficult to imagine the iPhone without its outside apps. This is a partial vindication of generative technologies over sterile ones, but with caveats that may make this the worst rather than the best of both worlds.

The application environment for the iPhone flips from the PC-with-antivirus-software's blacklist system to a whitelist scheme unchangeable by the user. With rare exceptions, such as a special ad hoc program allowing very limited distribution. Outside developers must register with Apple, promise not to disclose anything about how apps are written, and if approved they may then submit their software to Apple for review and possible inclusion in the iPhone App Store, the only way for the public at large to obtain it.

Apple can change its mind at any time about a particular piece of software's inclusion in its store. And even software already obtained from the iPhone App Store can be recalled—that's just a subset of Apple's ability to remotely reprogram any aspect of the phone at any time.

For vendors of iPhone apps, Apple's goodwill is thus vital. Apart from deciding whether an app lives or dies, Apple can feature favored apps in its store, and it can make app updates and bug fixes available slowly or quickly. For these reasons a gag order in the license agreement demanding that software authors not discuss codewriting for unreleased software was taken very seriously among app developers. Apple doesn't need to bring a lawsuit against a developer who violates license terms; it already has the power to destroy the iPhone-based livelihood of anyone disfavored, for any reason.

NetShare, an app that allowed users to gain wireless connectivity for their PCs through a connected iPhone, disappeared, and as of this writing the company says it doesn't know why. Box Office, an app that provided movie times, was removed from the iPhone App Store for several weeks. Its developer has declined any comment since its return. Other developers for mail and podcast-related programs say their apps were turned down with the explanation they were "duplicative" of (that is, competitive with) existing iPhone functionality.

Should we care? Apple likely wouldn't kill apps people really like since they make money along with the authors: 30% of all sales. And people think of an iPhone as a more unified device, expecting all of it to work at high quality, so gatekeeping might help keep malicious or poor-quality apps away.

On the other hand, people don't know what they're missing—and firms can be very ineffective, despite their own economic interests, in recognizing the value of truly novel contributions from outsiders that might take a while to catch on. Who would have invested in Wikipedia at the beginning? And if Wikipedia required an incumbent gatekeeper's approval or permission to get started, it might have failed to receive it—or languished at the bottom of a to-do list among hundreds of other apps and services awaiting review.

This phenomenon isn't exclusive to Apple, of course. Even today's PCs have a flavor of it. Microsoft offers a monthly malicious software removal tool, which unobtrusively goes through a PC to remove malware. Presumably it would become much less popular if Microsoft, or someone regulating Microsoft, tried to use the tool to remove software that people liked; no one seems to have tried to get Microsoft to kill anything yet, though, and such attempts are limited since any new app can immediately be installed on a PC—including one that shuts down a Microsoft app-removal tool.

Back on the whitelisting side of the spectrum, Web development platforms like Facebook Apps have restrictions that essentially mirror those of the iPhone. And when Facebook kills an app, the app is naturally not only unavailable to new users, but disabled for current ones, too. So Superwall or Secret Crush can go from millions of users to zero in a heartbeat. People learn about new apps through their friends' Facebook newsfeeds—and Facebook can adjust just how much news an app will generate there. A Great Apps program allows Facebook to pick winners and feature them more openly, even as some developers grumble that the functionalities they build are sometimes incorporated into apps written by Facebook itself[2]—and then effortlessly promoted more than the outsiders' original.

Its modest malicious software removal tool aside, imagine if Microsoft had adjusted Windows to act the way the iPhone and Facebook apps platforms do.

The iPhone apps model is powerful, and it is serving some useful purpose in shielding people, prospectively and retroactively, against bad code.

WordPerfect would owe Microsoft 30% on sales of every copy of its word processor—if it sold any, since Word could be featured by Microsoft to its users much more readily, or rejected entirely as duplicative of Word. (Recall that a main basis for the Microsoft antitrust case in the 1990s arose from Microsoft's attempts to force PC sellers to include the Internet Explorer browser on their PCs' desktops out of the box. The ways in which Facebook or Apple can feature their own apps over those of others dwarf that.) Of course, Microsoft could change that percentage owed at any time—or make it a flat fee. The makers of, say, Quicken, could find that they owe 70% or 80% on every app, take it or leave it. If they leave it, Quicken would stop working on every PC on which it had previously been installed.

And anyone objecting to an app—say, the movie and music industries beholding the rise of Kazaa or BitTorrent—could pressure Microsoft to kill it the day it appeared. We recently experienced this scenario when Hasbro, owner of the intellectual property rights to Scrabble in the U.S. and Canada, pressured Facebook to kill Scrabulous, a Scrabble knockoff. No court needed to weigh in on this decision.

We likely wouldn't accept this situation in PC architecture, and yet it is commonplace in the ecosystem that will soon replace it. Is the difference that Microsoft had overwhelming market share—an acknowledged monopoly—over PCs? That certainly counts, but even if one vendor doesn't capture the mobile phone or social networking spaces, the choices among them are shaping up to be choices among gated communities: equivalent to the old AOL vs. Prodigy vs. CompuServe, with the Internet not in the running. This is one reason why Google's Android project is so fascinating: an attempt to bring the generativity of the PC to the mobile phone space. Without a security model better than the PC's security provisions, however, Android is a tough proposition. How long will users tolerate a phone for which clicking on the wrong link can disable it?

The iPhone apps model is powerful, and it is serving some useful purpose in shielding people, prospectively and retroactively, against bad code. It is so powerful and popular that we will see it extended to PC-like platforms, too, with the 30-year run of open season for new software drawing to a close.

The way forward—for both PCs and smartphones—lies in a new security architecture that lets users make better-informed decisions about whether to run new software. We could aggregate data and make it freely available—how many experts have installed this same code? On average, what impact does the code have on the environment in which it runs, as measured by crashes or pop-ups or user satisfaction? A user deciding whether to run new code could use that data to make a simple decision, instead of letting the autocratic voice of preprogrammed security software dictate the result. Such an architecture would be more flexible than what we currently use. There are many details to work out, but without ways of managing our generative platforms without a central gatekeeper, chances seem strong that most people will accept—even demand—outside control.

References

1. Computers in use pass 1 billion mark: Gartner. Reuters (June 23, 2008); www.reuters.com/article/technologyNews/idUSL2324525420080623.
2. Stone, B. New tool from Facebook extends its Web presence. *New York Times* (July 24, 2008); www.nytimes.com/2008/07/24/technology/24facebook.html.
3. Zittrain, J. *The Future of the Internet—And How to Stop It.* Yale University Press, Penguin U.K., and Creative Commons, 2008; www.futureoftheinternet.org/download.

Jonathan Zittrain (zittrain@law.harvard.edu) is Professor of Law at Harvard Law School in Cambridge, MA.

DOI:10.1145/1435417.1435428

Alex E. Bell

From the Front Lines
Ground Control to Architect Tom...

Using David Bowie's classic song "Space Oddity" as a backdrop, the unscrupulous tactics of architecture astronauts are exposed.

PROJECT MANAGERS LOVE him, recent software engineering graduates bow to him, and he compels code warriors deep in the development trenches to wonder if a technology time warp may have passed them by. How can it be that no one else has ever proposed software development with the simplicity, innovation, and automation being trumpeted by Architect Tom? His ideas sound so space-aged, so futuristic. But why should that be so surprising? After all, Tom *is* an architecture astronaut![1] You see, architecture astronauts like Tom naturally think at such high levels of innovation because they spend much of their time in high orbit where low oxygen levels counterbalance the technological shackles imposed by the inconveniences of reality.

Little did I know it at the time, but my daughter inadvertently stumbled upon several foundational tenets of architecture astronautics years ago while watching a television cartoon show called "Princess Gwenevere and the Jewel Riders." One Saturday morning, while Gwenevere was feverishly fighting the forces of evil in the fairy-tale land of Avalon atop her beloved unicorn Sunstar, my daughter asked, "Daddy, are unicorns real?" I quickly debated the tactics to be taken to answer the question and decided that honesty and reality were going to be my guiding beacons. Unfortunately, my well-intended tactics had zero relevance in the ensuing discussion. Unicorns were real because my daughter wanted them to be and because I could offer no proof to suggest otherwise.

This seemingly innocuous exchange with my daughter, known as the "Sunstar Impasse of 1996," raised my awareness of a tactic that is most certainly described on page 1 of the official *Architecture Astronaut Handbook*. "An architecture astronaut in good standing shall exploit the impossibilities of proving nonexistence." For example, let's say that Architect Tom draws a picture of a spacecraft with an impressive array of dials and switches, describes its behavior with some complex yoo-mel diagrams, and then declares that the device's remaining production effort is merely an annoying detail. How does one prove that this spacecraft cannot be built when Architect Tom says that it can? Even worse, how does one convince nontechnical people that Architect Tom's spacecraft is heading for a fiery demise if there is a desperate reliance on his oxygen-challenged thinking to avoid controversy related to project commitments and promises?

Another foundational tenet most certainly included in the official *Architecture Astronaut Handbook* can be best illustrated by returning to the land of Avalon where Princess Gwenevere's fellow Jewel Riders, Tamara and Fallon, enter the scene with their unicorns and magical enchanted jewels in tow. On the occasions when my daughter invited friends over to watch the Jewel Riders with her, it was not unusual to overhear debate among them as to which of the Jewel Riders was the prettiest, which had the nicest unicorn, or which enchanted stone was the most desirable. Architecture astronauts similarly debate such fictional truths among themselves without the slightest concern that the foundations of their debates have no correlation with reality.

An example of an inane debate occurring between members of the Fraternal Brotherhood of Astronauts might involve one astronaut asking another, "Is it preferable to ride a unicorn or a magic carpet to work?" While the ensuing discussion would most certainly involve trading off the high costs of unicorn food versus the comfort of 200 knot-per-square-inch magic carpets, not a word would be spoken suggesting unicorns and flying carpets are make-believe. Speaking of similarly inane debate, I wonder which one of the Jewel Riders would be the most attracted to me if I were to travel to Avalon? My wife would undoubtedly suggest it would be the one clutching a white cane atop a seeing-eye unicorn.

Architecture astronauts are simple to recognize because they exhibit a number of signature qualities. One of the most prominent of those qualities is an uncanny ability to confidently speak on topics that sometimes momentarily humble even true subject-matter experts. Of course, a few probing questions are usually sufficient to expose even the most seasoned astronauts as the imposters they really are. Architect Tom and his co-orbiters further demonstrate astronautic allegiance by habitually speaking at extremely high levels of abstraction.[a] This tactic is very important because it allows astronauts to avoid engaging in tangible discussions that might lead to disclosing they really have no idea of what they are talking about. And finally, architecture astronauts often compromise themselves by voicing their dislike of "old" software technologies and design tactics. Astronauts prefer new technologies such as Web services, which now enable distributed computing, and XML, which has unharnessed the power of self-describing data.

At some point in the journey of a project being led by an architecture astronaut, the wheels on spacecraft begin to wobble and the spacesuit shows signs of transparency. It is usually a lack of tangible progress demonstrating the legitimacy of an astronautic vision that ultimately triggers a fiery descent from space. Beyond the traditional diversionary tactic of asserting that the design and development staffs are too incompetent to implement the vision, architecture astronauts rely heavily on the "spacewalk shuffle" where they move fast in their spaceboots to invent even further out-of-this-world ideas with which to divert and calm concerns with the original vision. The most clever architecture astronauts can string a project along all the way until its doom without ever being exposed as the charlatans they really are.

How is it that these astronauts can be placed in organizational positions with significant influence on the technical trajectory of a project? What qualifications entitle some of these astronauts to even have the title "architect" printed on their business cards or appended to their email signatures? The credential my wife earned in a food-handling course entitling her to sell nachos at our son's track meets represents more qualification for doing her job than many astronauts could produce for doing theirs. My wife, however, does not misrepresent herself as a nacho architect.

There is no evidence to suggest the Architecture Astronaut Continuum will cease to exist anytime soon, astronauts will continue to be hired, exposed, and fired. As a result, software projects will continue to be susceptible to the Continuum's accompanying carnage unless they adopt tactics to defend against it. The time has long passed for many software organizations to protect themselves from the damaging effects of architecture astronautics by requiring some form of credential or training before allowing their employees to call themselves architects or assume the responsibilities of one. The ability to create an illusion of knowledge among people who don't know any better is really a trait of a shrewd salesperson, not a software architect.

Are you an architecture astronaut that is diverting the technical efforts of your program from following a realistic and value-added trajectory? If so, fear not, there is hope. A flourishing segment of the global marketplace is being fueled by astronauts just like you who have recognized a need to get out of high orbit and restore the constraints of reality to their thinking. Flavors such as peppermint, bayberry, cranberry, and wintergreen are just a few that await at an oxygen bar near you!

Can you hear me Architect Tom?
Can you hear me Architect Tom?

The ability to create an illusion of knowledge among people who don't know any better is really a trait of a shrewd salesperson, not a software architect.

a Also noted by Joel Spolsky in 1.

Reference

1. Don't let architecture astronauts scare you. Joel on Software (Apr. 21, 2001); http://www.joelonsoftware.com/articles/fog0000000018.html.

Alex E. Bell (alex.e.bell@boeing.com) is a software architect with The Boeing Company.

DOI:10.1145/1435417.1435427
Phillip G. Armour

The Business of Software
The Ontology of Paper

The next generation of software engineering will involve designing systems without using paper-based formats, instead using software to develop software.

It is somewhat ironic that in our business we don't use much software to produce software. Excepting dynamic testing, we don't generally employ executable knowledge to create executable knowledge. We have compilers that convert a text form of instructions understandable to (some) humans into machine instructions executable by a machine. But a compiler doesn't know what we are *supposed* to do and it cannot generate or validate the external real-world functional knowledge. We have CASE tools and modeling tools that allow us to store some of the knowledge we need in a variety of formats such as directed graphs, tables, and dictionaries. But *we* have to put the knowledge into these tools—the tools cannot self-populate. And often, the only thing we can do with knowledge we have put into these tools is merely retrieve it and look at it.

Knowledge in most software development tools does not usually execute. A tool may verify that the knowledge format conforms to some conventional modeling rules, but few tools allow us to automatically validate that the contained knowledge is correct and applicable to the situation at hand in the real world.

Electronic Paper

As rich and clever (and expensive) as these tools may be, they mostly fail the test of executability. They are more like electronic paper than they are software. The real work of software development is done in word processors, list-format data catalogs, and simple constrained drawing tools. More correctly, most of the work in software development is done in the individual and collective brains of the developers—the word processors and other tools are the non-executable book-like places we put the knowledge once we have it.

An IDE Whose Time Has Not Quite Come

Integrated development environments (IDEs) have come a long way since the days of programming using the VI editor, but they are still mostly word processors with some specific look-up and retrieval processes and just a few executing functions such as condition checking (which, of course, we have to define).

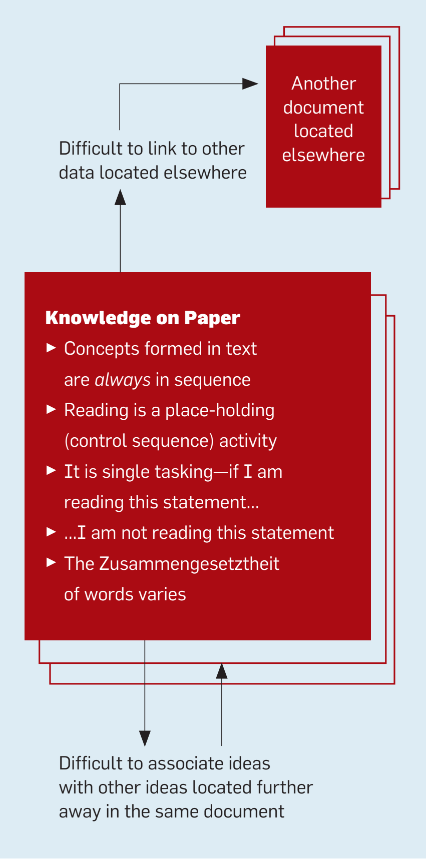
We spend most of our systems development effort collecting pieces of knowledge and putting them down on pieces of paper (electronic or otherwise) and it is interesting to consider just what the physical act of putting knowledge onto pieces of paper does to the knowledge (see the figure here).

Words are in sequence. When we put information into words on pieces of paper we apply a sequence. Words are in sequence. We can change the sequence words are in but in sequence words are nonetheless. So even if what we are describing is not sequential, when we put it in words, it will be.

Text is control oriented. The act of reading (and writing) words is a control-oriented task. There is always an associated point of attention—what is being read at that moment in time. The point of control generally moves from one statement to another according to the sequential behavior described previously. Putting ideas onto paper forces this control focus on the ideas we put on the paper.

Reading is single tasking. People do not usually read just one word at a time; they actually absorb patterns or blocks of several words at the same time. But our attention is mostly fixed

The ontology of paper.



at one place. If I am reading this word, I am not reading that word; if I am looking at this page, I am not looking at that page. Even if what we describe in words is concurrent, event-driven, and multi-tasking, once it is in words it will look like it is single-tasked.

Text has few connections. The only real relationship that exists between physical words is next-prior. This is the sequence of the words and it is tracked in a single-tasking mode using the place-holding reading control mechanism. We have developed different flavors of next-prior at different levels of abstraction to approximate different types of connections. We group sets of concepts into sentences and then into paragraphs and into chapters. We assign different purposes to different chapters to use the physical proximity to model the logical relationship. We indent paragraphs to show decomposition or ownership and there is usually some implicit relationship of knowledge elements at the same indentation level. If we want to reinforce a sequential relationship we may assign numbers to the paragraphs. If we want to de-emphasize sequence and infer equivalence, we may annotate the paragraphs with bullets.

In the electronic version of the written word we are able to embed hyperlinks to "join" concepts together that are physically separated in the document. But while the hyperlink allows physical connection of ideas and quicker transition between them, it doesn't clearly show proximity of the knowledge. On paper, the physical proximity of ideas is still the strongest mechanism for associating knowledge.

The systems we are building now are those for which the ontology of paper does not work very well and we have to rethink how we think about systems.

Words have limited scope. There is a relationship between the understandability of a document and the number of words. We manage this complexity by the mechanisms I've described and by adjusting the abstraction level of the words we use—we can use fewer words, but only if the meaning of these words (which is often located elsewhere) is more "knowledgeful." We see this clearly in certain disciplines such as the medical profession, which has created a domain-specific lexicon to allow more efficient communication. We have the option of using fewer and simpler words but then we inevitably convey less meaning since the sentences must contain less knowledge.

Insular association. Lastly, within one document, it is very difficult to associate ideas that are contained in another document. We do this through indexes and citations, but the ideas are still "further away." When we want to really bring them together we have to rewrite the ideas and include them in the source document (with attribution). This uses physical proximity of the words to emphasize the logical relationship of the ideas carried by the words.

Paper Is Good For…

If we were to use words and paper (in either the dead-wood or the electronic form), to describe the behavior of a system, we would expect it to work well for those systems where the knowledge maps closely to the ontology of the medium, specifically systems that are: sequential, single tasking, control based, mostly consisting of localized knowledge (the proximate knowledge is related and related knowledge is proximate), all the necessary knowledge is local and the abstraction level of system statements does not require extensive external reference because either the definition is simple or the complexity is abstracted into the language statements. And, of course, the knowledge would not be too "big" or too "complex."

Paper Is Not Good For…

On the other hand, we should expect that describing the operation of systems using paper formats would not work well for systems that are: multi-tasking, event-driven (especially out of state or time sequence), that need to interface with other knowledge repositories, especially where the abstraction level has not been defined and standardized, and that are large and complex.

If we look at the types of systems we were building a couple of decades ago, they fit well to the strengths of paper. They were silo'd, insular, control-based sequential systems that were, compared to today's systems, rather simple. But we don't build these kinds of systems much anymore. The systems we are building now are those for which the ontology of paper does not work very well and we have to rethink how we think about systems. Given that software is a knowledge medium, the future of software engineering clearly lies in constructing software artifacts using media, tools, and representational forms that:

- Can integrate knowledge both locally and remotely,
- Use different (and preferably programmable) representational forms than text that better lend themselves to the time- and state-based structure of the problem we are describing,
- Allow linking and manipulation of ideas remotely (for both machine operation and human understanding) and, most importantly…
- Are *executable*.

Steam Engines

It is a common cliché that we are currently going through an "Information Revolution" that may be more profound in its consequences than the development of steam engines was in the Industrial Revolution. This is undoubtedly true, but it is missing one important point: the Industrial Revolution did not occur when we built steam engines, it occurred when we used steam engines to build steam engines.

The true information and computing revolution will not occur until we use software to build software; until we really use executable knowledge to create executable knowledge.

And lose that paper.

Phillip G. Armour (armour@corvusintl.com) is a senior consultant at Corvus International Inc,. Deer Park, IL.

viewpoints

DOI:10.1145/1435417.1435429

Michael Cusumano

Technology Strategy and Management
The Legacy of Bill Gates

Assessing the pluses and minuses of the helmsmanship of Microsoft Corporation since its inception.

HALF A YEAR has now passed since Bill Gates retired from his day-to-day position at Microsoft last June. Gates remains chairman of the board and still spends about one day a week at the company. But the media marked his departure as a major event, and rightly so. Gates built and led, for more than 30 years, one of the most successful technology companies in history, and Microsoft's products touch nearly all of our lives every day. At the same time, Gates lives with the reputation that he was simply "lucky." Gates and Paul Allen founded Microsoft in 1975 to make programming languages for a new PC that had yet to come to market. Gates later got the IBM contract to produce DOS in 1980–1981 after first turning it down because he did not make operating systems. At least part of his legacy stems from this good fortune, especially since IBM was not clever enough to control the rights to the operating system (nor the microprocessor design, which Intel controlled) that Gates then sold to clone producers. Another view is that Gates was primarily a talented "hacker," with little or no management skills, and that Microsoft succeeded because of its growing market power stemming from DOS and then the Windows and Office monopolies.[a]

There is much truth to the argument that Gates and Microsoft were in the right place at the right time, but that fact does not explain why they were able to exploit the opportunities that appeared. In 1995, Richard Selby and I wrote in *Microsoft Secrets*: "Bill Gates of Microsoft may be the shrewdest entrepreneur and the most underrated manager in American industry today."[b] I still stand by this judgment, but now is a good time to reflect on what might be the ultimate legacy of Bill Gates.

On the positive side, Gates correctly foresaw the future when, in 1975, he dropped out of Harvard to co-found Microsoft based on the belief that every desktop would one day have a PC on it. He wanted those machines to run Microsoft software and he played a huge role in making this vision come true. Second, Gates recognized that hardware prices were decreasing and that most of the future value from computing would be in software. At the same time, he argued that users should pay for software so that programming could become a livelihood and software companies would have the money to invest in product development. The free software movement and the shift to services-driven business models are complicating the software business today but this strategy served Microsoft remarkably well for three decades.

a The best account of Gates and Microsoft through the early 1990s is Stephen Manes and Paul Andrews, *Gates: How Microsoft's Mogul Reinvented an Industry—and Made Himself the Richest Man in America*. Doubleday, New York, 1993.

b M.A. Cusumano and R.W. Selby, *Microsoft Secrets*. Free Press/Simon and Schuster, New York, 1995, 23.

Third, Gates correctly recognized the potential of the software products business, in contrast to the custom software, services, and maintenance business of the mainframe world. He realized that commonly used software programs could be standardized, packaged, and sold cheaply, like any other commodity. He was not the first to see this but he exploited the potential of shrink-wrapped software better than anyone else. Fourth, Gates early on understood the power of creating products that could become "platforms" (see my September 2008 column comparing Microsoft and Apple). Windows and Office continue to generate so much revenue because they are at the center of enormous ecosystems of users and complementors. Microsoft designs products that are more open than Apple and has worked much harder to cultivate partners. Fifth, as Selby and I also wrote in 1995, Gates stands out for how he, and the people he hired, uniquely combined an understanding of "the technology and the business." Simply being a great programmer did not land you a job at Microsoft. You had to understand how to make money from software as well as how to work in a team. Sixth, Gates led his company through at least two major technological transitions, either of which might have done in a less able leader and organization: from character-based to graphical computing, and from the desktop to the Internet. He also directed Microsoft's diversification from programming languages to desktop operating systems, applications, enterprise software, video games, and the MSN online business, among others.[c]

Finally, Gates has distinguished himself as a truly charismatic leader of a unique kind of company—a software company—which defies conventional rules of productivity and operations given that the marginal cost of reproducing a software product is essentially zero. Gates never had formal management training and displayed many rough edges in his first decade or two on the job. But, make no mistake about it. Gates learned how to manage, delegate, and lead thousands of extremely smart, talented people. He continues to set the standard for other high-technology entrepreneurs-turned-CEOs.

c My thanks to David Yoffie for reminding me of this last point.

On the negative side, first, part of the legacy must be that Gates himself was never very good at anticipating trends in the software business. His great strength was the ability to identify movements relatively early and then mobilize resources to exploit those trends. But he did not anticipate. This is true of graphical computing (led by Apple), Internet computing (led by Netscape), Internet search and other services (led by Yahoo, Google, and others), multi-platform computing (led by Sun Microsystems), open source computing (led by the Linux community and best exploited by Red Hat and IBM), the transition from products to services in the computer industry (foreseen best by IBM), software as a service (pioneered by Salesforce.com), or cloud computing (led by Amazon and Google). Second, Gates took too long to adjust to the position of power Microsoft attained and never quite understood the rules of competition differ for a firm with a monopoly market share (in this case, two monopolies). He testified reluctantly and weakly in the anti-trust trial. He never realized it was unnecessary as well as illegal for Microsoft to cross the antitrust line to battle competitors such as Netscape.[d]

Third, Gates let the company "get away" from him when it should not have, and he never demonstrated a "second wind." His early principles of how to manage a software company focused on hiring only top-notch people who understood both the technology and the business, and keeping development teams and products small in size. These principles started slipping in the mid-1990s. Microsoft reacted in a panic to Netscape and threw hundreds of programmers at building Internet Explorer and then dispatched thousands more to rebuild Windows and Office. Moreover, Gates was still chief software architect when Windows grew too large and complex in the early 2000s, tying too much functionality together to make every new feature an "integral part of Windows." Gates also kept the company tied to Windows when he could have decided that it was a platform company, whether the platform was Java or Linux or the Internet, and not just a Windows company. Microsoft has been recovering from the Windows quagmire and it has a new partnership with Novell to support Linux. But other managers have pushed these changes.

d See M.A. Cusumano and D.B. Yoffie. *Competing on Internet Time: Lessons from Netscape and Its Battle with Microsoft.* Free Press/Simon and Schuster, New York, 1998. Also, M.A. Cusumano, That's some fine mess you've made, Mr. Gates. *Wall Street Journal* (Apr. 5, 2000), A26.

Shareholders should be disappointed that Gates grew weary of being targeted for his success during the antitrust trial and started withdrawing from his responsibilities then, first relinquishing the CEO title to Steve Ballmer in 2000. While he took on the title of chief software architect, he spent increasing amounts of time and money on philanthropic ventures, as well as raising his new family. This is a natural and highly laudable transition for a person at his pinnacle to make. But we know from the case of Steve Jobs and Apple that it is not impossible for CEOs, even with families and outside interests, to reinvent themselves and reenergize the companies they lead. Steve Jobs has taken Apple to its greatest heights only in the last couple of years, with the iPod and the iPhone. Though Apple may never match Microsoft in revenues or profitability, Jobs and Apple clearly have achieved a thought leadership position in computing, digital services, and multimedia. Apple shareholders have benefitted greatly in the past few years while Microsoft's stock has languished or fallen. Moreover, Gates is departing just when Microsoft's business model of packaged software products is undergoing its most serious challenge since the Internet from free software and remotely delivered software as a service and "cloud computing," and even a new browser (or Web operating system?) from Google.

Perhaps it is unfair to end a discussion of Gates' legacy by describing what he was not or was not trying to do. What he did is more than enough for any career. No individual did more to grow the PC software business by bringing inexpensive, powerful software products to the masses. This achievement should be Gates' most enduring legacy. He will always be one of the greatest (and richest) entrepreneurs and managers ever produced by any country, and he is no longer underrated.

Michael Cusumano (cusumano@mit.edu) is Sloan Management Review Distinguished Professor of Management and Engineering Systems at the MIT Sloan School of Management and School of Engineering in Cambridge, MA.

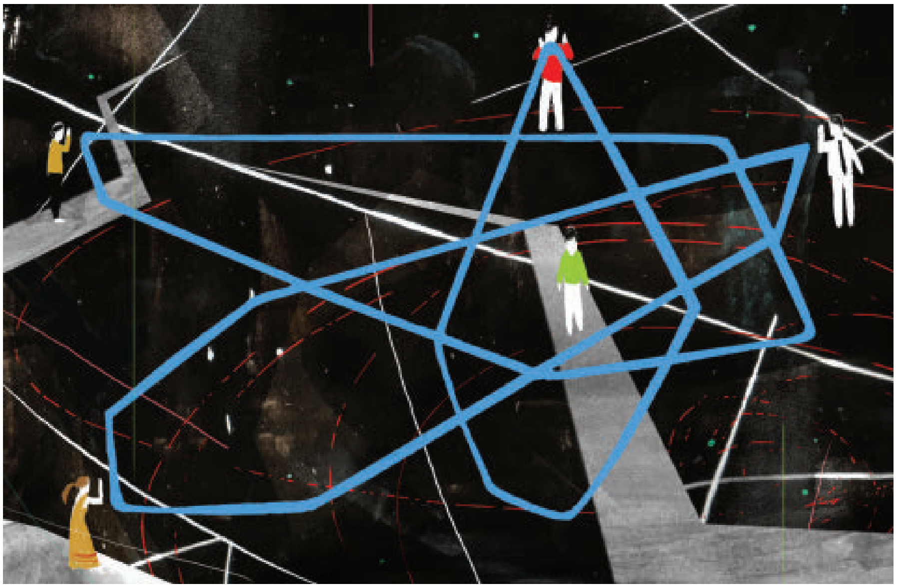
DOI:10.1145/1435417.1435430

Jon Crowcroft, S. Keshav, and Nick McKeown

Viewpoint
Scaling the Academic Publication Process to Internet Scale

A proposal to remedy problems in the reviewing process.

THE REVIEWING PROCESS for most computer science conferences originated in the pre-Internet era. In this process, authors submit papers that are anonymously reviewed by program committee (PC) members and their delegates. Reviews are typically single-blind: reviewers know the identity of the authors of a paper, but not vice versa. At the end of the review process, authors are informed of paper acceptance or rejection and are also given reviewer feedback and (usually) scores. Authors of accepted papers use the reviews to improve the paper for the final copy, and the authors of rejected papers use them to revise and resubmit them elsewhere, or withdraw them altogether.



Some conferences within the broader computer science community modify this process in one of three ways. With double-blind reviewing, reviewers do not know (or, at least, pretend not to know) the authors. With shepherding, a PC member ensures that authors of accepted papers with minor flaws make the revisions required by the PC. And, with rollover, papers that could not be accepted in one conference are automatically resubmitted to another, related conference.

Surprisingly, the advent of the Internet has scarcely changed this process. Everything proceeds as before, except that papers and reviews are submitted online or by email, and the paper discussion and selection process is conducted, in whole or in part, online.

ILLUSTRATION BY JON HAN

A naive observer, seeing the essential structure of the reviewing process preserved with such verisimilitude, may come to the conclusion that the process has achieved perfection, and that is why the Internet has had so little impact on it. Such an observer would be, sadly, rather mistaken.

Problems with the Current Review Process

We believe the paper review process suffers from at least five problems:

- A steady increase in the total number of papers: Because the number of experienced reviewers does not appear to be growing at the same rate, this has increased the average reviewer workload.
- Skimpy reviews: Some reviewers do a particularly poor job, giving numeric scores with no further justification.
- Declining paper quality: Although the best current papers are on par with the best papers of the past, we have found a perceptible decline in the quality of the average submitted paper.
- Favoritism: There is a distinct perception that papers authored by researchers with close ties to the PC are preferentially accepted with an implicit or overt tit-for-tat relationship.
- Overly negative reviews: Some people enjoy finding errors in other people's work. But this often results in reviews that are overly negative, disheartening beginner authors.

These problems are interrelated. The increase in the number of papers leads, at least partly, both to a decline in paper quality and a decline in the

Mechanism goals.

A1	Authors should not submit poor-quality papers
A2	Authors should become reviewers
R1	Reviewers should submit well-substantiated reviews
R2	Reviewers should not favor their friends
R3	Reviewers should not denigrate competing papers

quality of reviews. It also leads to an ever-increasing variance in paper quality. Similarly, as the acceptance rate of a conference declines, there is a greater incentive for reviewers to write overly negative reviews and favor their friends.

The Paper Publishing Game

Paper reviewing and publishing can be viewed as a game. There are three players in this game, who are assumed to be rational, in the usual economic sense, and who have the following incentives:

- Authors want to get published, or, at least, get detailed, but not necessarily positive, reviewer feedback on their work. They also don't want to be induced into becoming reviewers.
- Reviewers/PC members want to minimize their work (for instance, by giving scores, but no justifications), while trying to reject papers that compete with their own papers, and accepting papers from their friends. They want to reject unacceptable papers that would embarrass them. Finally, they want to get the prestige of being in the PC.
- Chairs/TCP/Research Community stakeholders want to have the highest quality slate of papers, while trying to include fresh ideas, and providing some sense of coverage of the field.

Interestingly, the problems outlined here arise because the existing paper reviewing process does not explicitly address these contradictory incentives. There is no explicit incentive for authors to become reviewers or for authors to limit the number of papers they submit, or to submit good-quality papers. There is no check on reviewers who write skimpy reviews,[a] are overly negative, or play favorites. No wonder the system barely works!

Mechanisms for Incentive Alignment

Our goals, illustrated in the table here, involve designing mechanisms such that it is incentive-compatible to do the right thing. Here, we describe some mechanisms to achieve these goals (correlated to the A1, A2, R1, R2, R3 labeling scheme established in the table). Our proposals include some steps that have been tried by some brave conference PC chairs. Others that are

a Other than a slight risk of embarrassment at the PC meeting.

novel and would need experimentation and experience.

Author Incentives. Our first mechanism addresses A1 using peer pressure. It requires the conference to publish not only the list of accepted papers, but also, for each author, the author's acceptance rate for that conference. For example, if an author were to submit two papers and none were accepted, the conference would report an acceptance rate of 0, and if one was accepted, the author would have an acceptance rate of 0.5. Because no author would like to be perceived to have a low acceptance ratio, we think this peer pressure will enforce A1.

Our second mechanism addresses A2 by raising the prestige of reviewing. For example, conferences can have a best reviewer award for the reviewer with the best review score[b] or give them a discount in the registration fee.

A more radical step would be to solve A1 and A2 simultaneously by means of a virtual economy, where tokens are paid for reviews, and spent to allow submission of papers.[c] Specifically, assuming each paper requires three reviews on average, reviewers are granted one token per review, independent of the conference, and the authors of a paper together pay three tokens to submit each paper. We recognize that this assumes all conferences expect the same level of reviewing: one could pervert this scheme by appropriate choice of reviewing venues. We ignore this fact for now, in the interests of simplicity. Continuing with our scheme, authors of accepted papers would be refunded one, two, or all their tokens depending on their review score. Authors of the top papers would therefore incur no cost, whereas authors of rejected papers would have spent all three of their tokens. Clearly, this scheme forces authors to become reviewers, and to be careful in using the tokens thus earned, solving A1 and A2.

We note that we obviously need to make tokens non-forgeable, non-replicable, and perhaps transferable. E-cash systems for achieving these goals are

The goal would be to have a standard way for members of the community to review and rank papers and authors both before and after publication.

well known[d]—they merely need to be adapted to a non-traditional purpose. We recognize that regulating the economy is not trivial. Over-damping the system would lead to conferences with too few papers, or too few reviewers. Underestimating the value of tokens would only slightly mitigate the current problems, but would add a lot of expensive overhead in the form of these mechanisms. Moreover, it is not clear how this system can be implemented. Indeed, even if it was, it would not be obvious how it can be bootstrapped, or whether it would have unintended consequences. One possible technique would be to start by publishing signed reviews and rely on technologies such as Citeseer and Google Scholar as we describe here in more detail.

Reviewer Incentives. We first discuss dealing with R1 and R3. We propose that authors should rate the reviews they receive for their papers, while preserving reviewer confidentiality. Average (non-anonymized) reviewer scores would then be circulated among the PC. No PC member wants to look bad in front of his or her peers, so peer pressure should enforce R1 and R3 (PC collusion will damage the conference reputation). Note that we expect most authors to rate detailed but unfavorable reviews highly.

An even more radical alternative is for reviews to be openly published with the name of the reviewer. The idea is that reviewers who are not willing to publish a review about a paper

b See the subsection *Reviewer Incentives* for details on review scoring.

c We have been informed that this scheme was first suggested by Jim Gray, though we cannot find a citation to this work.

d For example, David Chaum's seminal work "Blind signatures for untraceable payments," *Advances in Cryptology Crypto '82*, Springer-Verlag (1983), 199–203.

We believe the academic community as a whole desires such a system. However, we also realize such a system can also be subverted.

are perhaps inherently conflicted and therefore should not be reviewing that paper. Of course, there is a danger that public reviews will be too polite, but this will no doubt sort itself out over time. The advantage of using true identities (verinyms) is that this handles R1, R2, and R3. Alternatively, reviews could be signed with pseudonyms, where the pseudonyms could persist across conferences. Nonce pseudonyms will protect the nervous but prevent building reputation. There is a fundamental balance between anonymity and credibility that we cannot hope to solve.

A Grand Unified Mechanism

A deeper examination of the incentive structure suggests that perhaps the real problem is that too much of the work of submitting and selecting papers is hidden. What if the entire process were made open, transparent, and centralized? The goal would be to have a standard way for members of the community to review and rank papers and authors both before and after publication, in a sense adding eBay-style reputations to Google Scholar or arXiv. All papers and reviews would be public and signed, with either pseudonyms or verinyms. This system, would, in one fell swoop achieve many simultaneous goals:

- Readers can draw their own conclusions (and tell the world) about the quality of papers published by an author. This would encourage authors not to submit bad papers (achieving A1).
- Community members who publish often and review rarely would be exposed, achieving A2.
- We would see the reviews and the names of the reviewers alongside the paper, addressing R1, R2, and R3.
- We get to see whose opinions correlate well with our own to help decide what papers to read.
- There is a good chance that very good papers that end up as technical reports or in smaller, less well known conferences, are raised to the top by popular acclaim.
- The system would allow continued discussion and feedback about papers even after they have been published (1) to help others (busy senior people, and new people not knowing where to start), and (2) to provide an opportunity for others to participate in the discussion and debate.

We believe the academic community as a whole desires such a system. However, we also realize such a system can also be subverted. As with e-cash, the hardening of reputation systems to resist collusion and other attacks is well known, and we merely need to import the appropriate machinery and techniques.

Conclusion

We have identified the underlying incentive structure in the paper publishing process and shown where these incentives lead to poor outcomes. These insights allow us to propose several mechanisms that give incentives to authors, reviewers, and the community to do the "right thing." We accept that there has been much altruism in the past, but in today's resource-scarce world, it may not be fair to rely on this any longer. We recognize our work is preliminary and leaves out many important details but nevertheless hope these ideas will serve as the foundation of a fundamental rethinking of the process. We hope at least some of our proposals will make their way into future conferences, workshops, and publications. C

Jon Crowcroft (jon.crowcroft@cl.cam.ac.uk) is the Marconi Professor of Communications Systems at the University of Cambridge, Cambridge, U.K.

S. Keshav (keshav@uwaterloo.ca) is a professor and Canada Research Chair in Tetherless Computing in the Cheriton School of Computer Science University of Waterloo, Ontario, Canada.

Nick McKeown (nickm@stanford.edu) is an associate professor of electrical engineering and computer science at Stanford University, Stanford, CA.

An earlier version of this material was published in *Proceedings of the Workshop on Organizing Workshops, Conferences, and Symposia for Computer Systems (WOWCS 2008).*

At a time when IT industries, global economies, and research funding operate in fits and starts, it is clear that computing professionals find enduring value in what it means to be an ACM member.

DOI:10.1145/1435417.1435425

Stuart I. Feldman

ACM's Annual Report

I'm a long-time member (40 years!) of ACM. As such, I've come to expect great things from this organization, having witnessed over these many years how expertly the Association's leadership, volunteers, and staff have worked to advance and inspire the discipline of computer science through its array of publications, conferences, educational initiatives, and professional development services. The ACM offerings have grown with the scope and importance of the computing field. But I would be hard-pressed to recall in my four decades a time as eventful and exceptional in these causes (and *more*) as recorded by ACM in FY08.

In a 12-month period, the Association saw both professional and student memberships reach an all-time high; a completely revitalized flagship magazine launch to rave reviews; our international initiatives result in a new ACM China office and a multiyear plan for raising the Association's visibility in India; and ACM's Digital Library provide offerings that have never been more plentiful nor its features more user-friendly.

The report that follows will detail these accomplishments, as well as many others, that have occurred in an extraordinary short timeframe. Perhaps the best indicator of the success of these efforts is reflected in the membership statistics. By the end of the fiscal year, ACM membership stood at a record 89,000. Indeed, professional memberships saw the largest year-over-year net increase in over 20 years. Moreover, student membership reached a milestone at 22,500. At a time when IT industries, global economies, and research funding operate in fits and starts, it is clear that computing professionals (present and future) find enduring value in what it means to be an ACM member.

Moreover, major corporations continue to support ACM's keen ability in recognizing technical excellence by sponsoring or greatly increasing the cash value of a number of the Association's prestigious awards. Of particular note this year was the first-ever ACM-Infosys Foundation Award in Computing Sciences, recognizing contributions by young scientists and system developers to a contemporary innovation.

As there are achievements to laud, there are as many challenges to be met. ACM must continue its global outreach, fostering critical international relationships and bringing ACM products and services to new and different computing audiences. And as a growing number of students question the future of a computing career, ACM remains committed to understanding and addressing the deep issues related to the image of computing and the health of the discipline and profession. ACM is partnering with several organizations, building task forces, and creating initiatives to raise awareness and promote the possibilities offered by the field.

As we look to the future, our opportunities have never been greater and, with the ongoing support of our dedicated volunteers, members, and industry partnerships, we will realize them. As this year has taught us, ACM is capable of surprising even those of us who thought we had seen it all.

It was an honor to serve as your president.

Stuart I. Feldman, ACM PRESIDENT

ACM's Annual Report for FY08

ACM, the Association for Computing Machinery, is an international scientific and educational organization dedicated to advancing the arts, sciences, and applications of information technology.

Publications

The centerpiece of the ACM Publications portfolio is the ACM Digital Library and its associated *Guide to Computing Literature*. At the end of FY08 there were more than two million pages of full-text articles in the DL, and more than 1,196,000 bibliographic citations in the *Guide*. Indeed, nearly 18,000 articles were loaded in the DL this year, and more than 86,000 citations were added to the *Guide*. Moreover, in April ACM introduced several major features to the DL Library, including detailed Author Profile pages as well as citation and usage statistics that provide a snapshot of an individual author's contributions to computing, and some measure of their influence in advancing the field.

After months of extensive planning and preparation, the newly revitalized *Communications of the ACM* debuted at the end of FY08. ACM's flagship now boasts a stellar international editorial board charged with identifying and shepherding its editorial content; committed to bringing to the table each month the latest news, practical applications, industry trends, cutting-edge research; thought-provoking commentary of the highest quality and professional value. Along with a classic and clean redesign, the magazine now hosts six editorial sections (News, Viewpoints, Practice, Contributed Articles, Review Articles, and Research Highlights) along with a new Web component to debut shortly.

Three new periodicals debuted this year: *Transactions on Reconfigurable Technology and Systems*; *Transactions on Accessible Computing*; and *Journal on Computing and Cultural Heritage*. In addition, the ACM Publications Board approved the proposal for an *ACM Transactions on Computational Theory*.

Education

The ACM Education Board finished a prolific year filled with projects and initiatives designed to reverse declining enrollments in computing disciplines and increase ACM's visibility within the worldwide educational community. The Board continues to analyze the dynamics of the enrollment downturn and find creative ways to foster a positive image of computing among young people, working closely with the Computer Science Teachers Association (CSTA) in supporting K–12 computing efforts. They also took major steps in updating the computing curricula guidelines in computer science, information systems, and information technology as well as established a master's-level initiative to evaluate the effectiveness of the degree on a global scale.

The CSTA took on several projects over the past year to promote and support the teaching of computer science and other computing disciplines at the K–12 level. The group, along with SIGCSE, organized a "Roadshow" aimed at helping colleges and universities improve their K–12 outreach efforts. Several other CSTA-led workshops and symposia focused on curriculum and leadership initiatives for educators throughout the U.S. The group also spirits international efforts, this year working with Pacific Rim educators as well as the Israeli Teachers Association.

The NSF awarded a grant to WGBH (the Boston-based PBS station), as lead investigator, and ACM, as co-principal investigator, to reshape the image of computing. The $850,000 two-year project will study the images and perceptions of computing held by high school students, particularly Latina girls and African American boys. The goal is to develop and rigorously test messages that would be effective in shifting negative perceptions and create strategies through these messages

ACM Council

PRESIDENT
Stuart I. Feldman

VICE PRESIDENT
Wendy Hall

SECRETARY/TREASURER
Alain Chesnais

PAST PRESIDENT
David A. Patterson

SIG GOVERNING BOARD CHAIR
Joseph A. Konstan

PUBLICATIONS BOARD CO-CHAIRS
Ronald Boisvert, Holly Rushmeier

MEMBERS-AT-LARGE
Michel Beaudouin-Lafon, Chuang Lin, Bruce Maggs, Barbara Ryder, Kevin Scott, Jeannette Wing, David S. Wise

SGB COUNCIL REPRESENTATIVES
Norman Jouppi
Robert A. Walker
Alexander Wolf

ACM Headquarters

EXECUTIVE DIRECTOR/CEO
John R. White

DEPUTY EXECUTIVE DIRECTOR/COO
Patricia M. Ryan

2007 ACM Award Recipients

A.M. TURING AWARD
Edmund M. Clarke, E. Allen Emerson, Joseph Sifakis

ACM-INFOSYS FOUNDATION AWARD IN THE COMPUTING SCIENCES
Daphne Koller

ACM/AAAI ALLEN NEWELL AWARD
Leonidas J. Guibas

GRACE MURRAY HOPPER AWARD
Vern Paxson

ACM-IEEE CS ECKERT-MAUCHLY AWARD
Mateo Valero
Karl V. Karlstrom

OUTSTANDING EDUCATOR AWARD
Randy Pausch

EUGENE L. LAWLER AWARD FOR HUMANITARIAN CONTRIBUTIONS WITHIN COMPUTER SCIENCE AND INFORMATICS
Randolph Y. Wang

OUTSTANDING CONTRIBUTION TO ACM AWARD
Robert A. Walker

DISTINGUISHED SERVICE AWARD
David A. Patterson

PARIS KANELLAKIS THEORY AND PRACTICE AWARD
Bruno Buchberger

SIAM/ACM PRIZE IN COMPUTATIONAL SCIENCE AND ENGINEERING AWARD
Chi-Wang Shu

SOFTWARE SYSTEM AWARD
David Harel, Hagi Lachover, Amnon Naamad, Amir Pnueli, Michal Politi, Rivi Sherman, Mark Trakhtenbrot, Aron Trauring

ACM PRESIDENTIAL AWARD
Stephen R. Bourne, Patricia M. Ryan, Barbara G. Ryder, Moshe Y. Vardi

DOCTORAL DISSERTATION AWARD
Sergey Yekhanin

HONORABLE MENTION
Vincent Conitzer, Yan Liu

that will eventually build a potential national campaign to draw interest and young students into the field.

Professional Development

The Professions Board transformed the *Queue* Web site (queue@acm.org), moving the publication from a print delivery model to an online magazine. With "digital *Queue*" as its foundation, the Board oversaw the specifications and implementation of the first phase of the Web site for practitioners both within and outside ACM complete with capabilities designed to encourage community input and collaboration. The *Queue* Editorial Board also joined the efforts to revitalize *Communications* this year by overseeing the monthly Practice section of the magazine and providing articles of great interest to today's practitioners.

ACM relaunched the Career & Job Center last fall, greatly expanding its scope of job opportunities in the computing industry. In partnership with Job Target, the site now offers exclusive career-building features, including access to hundreds of corporate job postings, résumé posting, an advanced Job Alert system, and live career advice available to assist in job-seeking preparations.

ACM further expanded its Online Books and Courses site (http://pd.acm.org/); a highly popular resource among the membership for its valuable materials designed to polish technical and professional skills. Over 9,000 members have taken advantage of the 2,200 Skillsoft online technology and business courses now available as well as the 1,100 free online e-books from Safari and Books24X7. This resource is offered to professional members as well as graduate students.

The partnership between SIGACCESS and the National Alliance for Access to Computing Careers (AccessComputing) continues to flourish. The goal of this combined effort is to increase the representation of people with disabilities in a wide range of computing careers, including CS, IT, and computer engineering.

Public Policy

The U.S. Public Policy Committee of ACM (USACM) had an extraordinary year in raising the visibility and influence of ACM with respect to U.S. public policy, meeting with policy-making groups, including Congress, the Federal Trade Commission, and the Election Assistance Commission. In the last fiscal year, members have cautioned Congress against filtering technologies to deal with copyright infringement; issued statements on the REAL ID rules; Web accessibility, and met with policymakers to promote e-voting standards. USACM also works closely with the Computing Research Association on issues that impact the computing field.

The new ACM Education Policy Committee (ACM EPC), established to educate policymakers about the role of computer science in the K–12 system, held its first meeting this year to focus its priorities. The committee has already engaged the National Science Foundation to improve research opportunities in CS education at the grade-school level. In addition, EPC has initiated discussions with leaders of the National Council of Teachers of Mathematics and has successfully partnered with Achieve.org—a high-profile education organization led by U.S. governors and corporate leaders—to include a CS course in its Advanced Diploma Project.

The ACM Committee on Computers and Public Policy continues to represent ACM interests with respect to a variety of internationally relevant issues pertaining to computers and public policy. In the last year, members of the committee have participated in panels and workshops examining e-voting technologies, election integrity, cyber security, and system trustworthiness.

Major inroads were made this year to expand ACM's internationalization initiative. Paramount in this movement was the opening of an ACM office in China.

ACM's commitment to the issue of women in computing was apparent this year with the decision to elevate the ACM-W Committee to ACM-W Council (AWC). The AWC will be a part of all leadership discussions within ACM and is chartered to increase the awareness of, and interest in, the issue of gender diversity across all ACM activities.

Students

The 32nd Annual International Collegiate Programming Contest World Finals proved a global media magnet. From 6,700 teams representing 1,821 universities in 83 countries competing at 213 sites from September to December, 100 teams advanced to the World Finals last April in Banff, Alberta, Canada. Of the top 10 teams at the 2008 ICPC, four were from Russian universities and three teams represented universities in North America. The top spot went to St. Petersburg University of Information Technology, Mechanics and Optics (Russia).

The ACM Student Research Competition, sponsored by Microsoft Research, continues to offer a unique forum for undergraduate and graduate students to present original research at well-known ACM-sponsored and co-sponsored conferences before a panel of judges and attendees. This year's winners hailed from Purdue University, Colorado State, University of Tennessee, City College of New York, IIT Bombay, and the University of Waterloo.

ACM's Committee on Women in Computing initiated a program to provide support for undergraduate and graduate women students in computer science programs who are interested in attending research conferences. Up to 12 scholarships will be awarded annually; and high school students will also be considered for conference support.

SIGCOMM instituted a "Rising Star" award, recognizing a young researcher (no older than 35) who has made outstanding research contributions to the field of communications networks during the early part of his or her career.

Conferences

SIGGRAPH attracted almost 25,000 artists, researchers, gaming experts, filmmakers, and developers representing 79 countries to its annual conference. The San Diego-based meeting also drew 230 companies to its exhibition hall, an increase of 12% over the previous year.

SIGMM reported a record number of attendees to its annual ACM Multimedia Conference held in Augsburg, Germany. The meeting also attracted a record number of paper submissions (up 43% from 2006) and offered a greater variety of content tracks than ever before.

SIGMOBILE experimented with co-locating its annual conference on mobile computing and networking with its annual symposium on mobile ad hoc networking and computing. Attendees reacted so favorably to the experiment, the SIG is planning try it again in 2010.

Local Activities

The Membership Services Board chartered 47 new chapters in FY08, six of which were international professional chapters. Of the 40 new student chapters, 20 were internationally based.

The ACM History Committee continues to foster collection, preservation, and interpretation of ACM's history and its role in the development of computing and information technology. An additional six oral histories with ACM luminaries were completed this year. More progress was also made on the committee's Web site (http://history.acm.org) with the goal of assembling a critical mass of source historical material that will enable historians to analyze and organize a rich history of ACM.

International

Major inroads were made this year to expand ACM's internationalization initiative. Paramount in this movement was the opening of an ACM office in China in December 2007. The office, hosted by Tsinghua University, supports several functions including maintaining the ACM China Web site and actively promoting ACM membership within the Chinese academic community. Indeed, over 300 members joined ACM through the China office in its first six months of operation.

The ACM India Task Force met to address how the Association could best support the discipline and field of computing in India and in the process be more relevant to, and visible within, the Indian computing community. A number of key issues and potential directions surfaced from that meeting, including ACM's strengths in engaging India's CS education at the university level as well as the unique challenges of and potential opportunities for tapping into India's computing community.

A meeting with European computing leaders last May helped ACM strengthen its ties with the region and better understand the key issues and initiatives within the European academic, research, and professional computing communities.

The Education Board continues to expand its global reach, particularly throughout Europe and Asia. The Board participated in activities associated with Informatics Europe as well as European accreditation efforts. Board members helped plan the recent Education Summit in China, and kept a close eye on the activities leading to

Balance Sheet: June 30, 2008 (in Thousands)

ASSETS	
Cash and cash equivalents	$20,783
Investments	49,045
Accounts receivable and other current assets	6,344
Deferred conference expenses and other assets	5,626
Fixed assets, net of accumulated depreciation & amortization	1,621
Total Assets	**$83,419**
LIABILITIES AND NET ASSETS	
Liabilities:	
Accounts payable, accrued expenses, and other liabilities	$7,396
Unearned conference, membership, and subscription revenue	19,061
Total liabilities	**$26,457**
Net assets:	
Unrestricted	51,343
Temporarily restricted	5,619
Total net assets	**56,962**
Total liabilities and net assets	**$83,419**
Optional contributions fund – program expense	($000)
Education board accreditation	$50
USACM Committee	20
Total expenses	**$70**

Statement of Activities: Year ended June 30, 2208 (in Thousands)

REVENUE	Unrestricted Net Assets	Temporarily Restricted Net Assets	Total
Membership dues	$8,873		$8,873
Publications	15,148		15,148
Conferences and other meetings	25,277		25,277
Interests and dividends	2,365		2,365
Net (depreciation) of investments	(1,570)		(1,570)
Contributions and grants	3,046	$1,084	4,130
Other revenue	391		391
Net assets released from restrictions	923	(923)	0
Total Assets	**54,453**	**161**	**54,614**
EXPENSES			
Program:			
Membership processing and services	$868		$868
Publications	11,274		11,274
Conferences and other meetings	22,124		22,124
Program support and other	7,313		7,313
Total	**41,579**		**41,579**
Supporting services:			
General administration	8,891		8,891
Marketing	1,436		1,436
Total expenses	**51,906**		**51,906**
Increase (decrease) in net assets	2,547	161	2,708
Net assets at the beginning of the year	48,796	5,458	54,254
Net assets at the end of the year	**$51,343***	**$5,619**	**$56,962***

* Includes SIG Fund balance of $28,142K

the signing of the Seoul Accord, created to establish a system of substantial equivalency of educational programs in the computing and IT-related disciplines that will lead to enhanced mobility of professionals in the field.

ACM's Special Interest Groups continued their international initiatives in FY08, reflecting their pledge to expand their professional outreach. Among the SIGs making global connections in FY08 were: SIGCSE's 2008 Conference on Innovation and Technology in Computer Science Education held in Madrid; SAC 2008 held in Brazil; SIGGRAPH newly chartered professional chapters in Shanghai and Madrid; SIGMICRO held its annual Computing Frontiers conference in Ischia, Italy and its CASES conference in Salzburg, Austria; SIGSAC's Symposium on Information, Computer and Communications Security was held in Tokyo; and SIGWEB's hypertext conference in Manchester, England, and its information and knowledge management conference in Lisbon, Portugal.

Electronic Community

Over the past year, the Computer Science Teachers Association has built an impressive collection of podcasts for interviews done with key thinkers and educators in the CS community. The collection now counts 32 podcasts with more in production.

The Distinguished Speakers Program launched a new Web site (www.dsp.acm.org) providing a sounder look-and-feel and better functionality.

SIGIR is driving a project to preserve their historical documents, including books, journals, and conference papers important to the IR community. The project will capture valuable resources to the community that are deteriorating in paper form as well as extinct digital media.

SIGCOMM's Web site was revised and reorganized, adopting a content management system for easier and more consistent updates, better navigation, an events calendar, and news.

Recognition

The first ACM-Infosys Foundation Award in the Computing Sciences, recognizing young scientists whose innovations are having a dramatic impact on the computing field, debuted last summer, carrying a prize of $150,000 endowed by the Infosys Foundation.

The ACM Fellows Program, established in 1993 to honor outstanding ACM members for their achievements in computer science and IT, inducted 38 new fellows in FY08, bringing the total count to 632.

ACM's advanced member grades were also announced. The Association recognized 171 Senior Members in FY08 who have at least 10 years of professional experience, five years of ACM professional membership, and have demonstrated performance and accomplishment that set them apart from their peers. ACM also recognized seven Distinguished Engineers and 13 Distinguished Scientists—all having the distinction of at least 15 years of professional experience and who have demonstrated significant accomplishments or made a significant impact on the computing field.

SIGMOD renamed its annual doctoral dissertation award this year in honor of Jim Gray, a revered member of the database community and a luminary in the computing field.

ACM, Uniting the World's Computing Professionals, Researchers, Educators, and Students

Dear Colleague,

At a time when computing is at the center of the growing demand for technology jobs worldwide, ACM is continuing its work on initiatives to help computing professionals stay competitive in the global community. ACM's increasing involvement in initiatives aimed at ensuring the health of the computing discipline and profession serve to help ACM reach its full potential as a global and diverse society which continues to serve new and unique opportunities for its members.

As part of ACM's overall mission to advance computing as a science and a profession, our invaluable member benefits are designed to help you achieve success by providing you with the resources you need to advance your career and stay at the forefront of the latest technologies.

MEMBER BENEFITS INCLUDE:

- Access to ACM's **Career & Job Center** offering a host of exclusive career-enhancing benefits
- **Free e-mentoring services** provided by MentorNet®
- **Full access to over 3,000 online courses** from SkillSoft®
- **Full access to 600 online books** from Safari® Books Online, featuring leading publishers, including O'Reilly (Professional Members only)
- **Full access to 500 online books** from Books24x7®
- A subscription to ACM's flagship monthly magazine, ***Communications of the ACM***
- Full member access to the new ***ACM Queue*** website featuring blogs, online discussions and debates, plus video and audio content
- The option to subscribe to the full **ACM Digital Library**
- The **Guide to Computing Literature**, with over one million searchable bibliographic citations
- The option to connect with the **best thinkers in computing** by joining **34 Special Interest Groups** or **hundreds of local chapters**
- **ACM's 40+ journals and magazines** at special member-only rates
- ***TechNews***, ACM's tri-weekly email digest delivering stories on the latest IT news
- ***CareerNews***, ACM's bi-monthly email digest providing career-related topics
- ***MemberNet***, ACM's e-newsletter, covering ACM people and activities
- **Email forwarding service & filtering service**, providing members with a free acm.org email address and **Postini** spam filtering
- And much, much more

ACM's worldwide network of over 92,000 members range from students to seasoned professionals and includes many of the leaders in the field. ACM members get access to this network and the advantages that come from their expertise to keep you at the forefront of the technology world.

Please take a moment to consider the value of an ACM membership for your career and your future in the dynamic computing profession.

Sincerely,

Wendy Hall

President
Association for Computing Machinery

Association for Computing Machinery

Advancing Computing as a Science & Profession

Association for Computing Machinery

Advancing Computing as a Science & Profession

membership application & *digital library* order form

Priority Code: ACACM28

You can join ACM in several easy ways:

Online	Phone	Fax
http://www.acm.org/join	*+1-800-342-6626 (US & Canada)* *+1-212-626-0500 (Global)*	*+1-212-944-1318*

Or, complete this application and return with payment via postal mail

Special rates for residents of developing countries:
http://www.acm.org/membership/L2-3/

Special rates for members of sister societies:
http://www.acm.org/membership/dues.html

Please print clearly

Name

Address

City | State/Province | Postal code/Zip

Country | E-mail address

Area code & Daytime phone | Fax | Member number, if applicable

Purposes of ACM

ACM is dedicated to:
1) advancing the art, science, engineering, and application of information technology
2) fostering the open interchange of information to serve both professionals and the public
3) promoting the highest professional and ethics standards

I agree with the Purposes of ACM:

Signature

ACM Code of Ethics:
http://www.acm.org/serving/ethics.html

choose one membership option:

PROFESSIONAL MEMBERSHIP:

- ❑ **ACM Professional Membership: $99 USD**
- ❑ **ACM Professional Membership plus the ACM Digital Library: $198 USD ($99 dues + $99 DL)**
- ❑ **ACM Digital Library: $99 USD (must be an ACM member)**

STUDENT MEMBERSHIP:

- ❑ **ACM Student Membership: $19 USD**
- ❑ **ACM Student Membership plus the ACM Digital Library: $42 USD**
- ❑ **ACM Student Membership PLUS Print *CACM* Magazine: $42 USD**
- ❑ **ACM Student Membership w/Digital Library PLUS Print *CACM* Magazine: $62 USD**

All new ACM members will receive an ACM membership card.
For more information, please visit us at www.acm.org

Professional membership dues include $40 toward a subscription to *Communications of the ACM*. Member dues, subscriptions, and optional contributions are tax-deductible under certain circumstances. Please consult with your tax advisor.

RETURN COMPLETED APPLICATION TO:

Association for Computing Machinery, Inc.
General Post Office
P.O. Box 30777
New York, NY 10087-0777

Questions? E-mail us at acmhelp@acm.org
Or call +1-800-342-6626 to speak to a live representative

Satisfaction Guaranteed!

payment:

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc. in US dollars or foreign currency at current exchange rate.

❑ Visa/MasterCard ❑ American Express ❑ Check/money order

❑ Professional Member Dues ($99 or $198)	$ ____________
❑ ACM Digital Library ($99)	$ ____________
❑ Student Member Dues ($19, $42, or $62)	$ ____________
Total Amount Due	$ ____________

Card # | Expiration date

Signature

DOI:10.1145/1435417.1435432

Building reliable distributed systems at a worldwide scale demands trade-offs between consistency and availability.

BY WERNER VOGELS

Eventually Consistent

AT THE FOUNDATION of Amazon's cloud computing are infrastructure services such as Amazon's S3 (Simple Storage Service), SimpleDB, and EC2 (Elastic Compute Cloud) that provide the resources for constructing Internet-scale computing platforms and a great variety of applications. The requirements placed on these infrastructure services are very strict; they need to score high marks in the areas of security, scalability, availability, performance, and cost-effectiveness, and they need to meet these requirements while serving millions of customers around the globe, continuously.

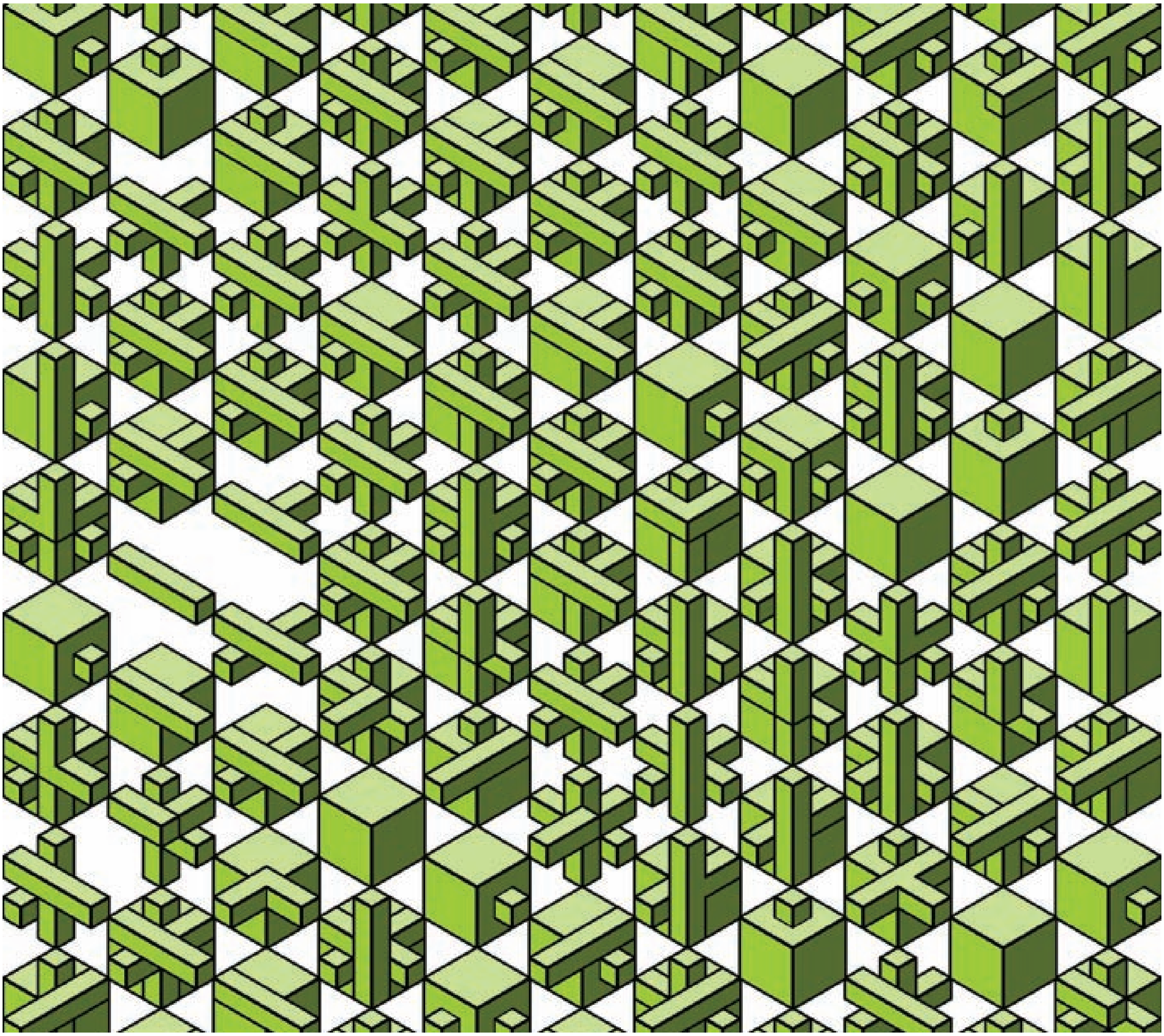
Under the covers these services are massive distributed systems that operate on a worldwide scale. This scale creates additional challenges, because when a system processes trillions and trillions of requests, events that normally have a low probability of occurrence are now guaranteed to happen and must be accounted for upfront in the design and architecture of the system. Given the worldwide scope of these systems, we use replication techniques ubiquitously to guarantee consistent performance and high availability. Although replication brings us closer to our goals, it cannot achieve them in a perfectly transparent manner; under a number of conditions the customers of these services will be confronted with the consequences of using replication techniques inside the services.

One of the ways in which this manifests itself is in the type of data consistency that is provided, particularly when many widespread distributed systems provide an *eventual consistency* model in the context of data replication. When designing these large-scale systems at Amazon, we use a set of guiding principles and abstractions related to large-scale data replication and focus on the trade-offs between high availability and data consistency. Here, I present some of the relevant background that has informed our approach to delivering reliable distributed systems that must operate on a global scale. (An earlier version of this article appeared as a posting on the "All Things Distributed" Weblog and was greatly improved with the help of its readers.)

Historical Perspective

In an ideal world there would be only one consistency model: when an update is made all observers would see that update. The first time this surfaced as difficult to achieve was in the database systems of the late 1970s. The best "period piece" on this topic is "Notes on Distributed Databases" by Bruce Lindsay et al.[5] It lays out the fundamental principles for database replication and discusses a number of techniques that deal with achieving consistency. Many of these techniques try to achieve *distribution transparency*—that is, to the user of the system it appears as if there is only one system instead of a number of collaborating systems. Many systems during this time took the approach that it was better to fail the complete system than to break this transparency.[2]

In the mid-1990s, with the rise of larger Internet systems, these practices were revisited. At that time people began to consider the idea that availability was perhaps the most impor-



tant property of these systems, but they were struggling with what it should be traded off against. Eric Brewer, systems professor at the University of California, Berkeley, and at that time head of Inktomi, brought the different tradeoffs together in a keynote address to the Principles of Distributed Computing (PODC) conference in 2000.[1] He presented the *CAP theorem*, which states that of three properties of shared-data systems—data consistency, system availability, and tolerance to network partition—only two can be achieved at any given time. A more formal confirmation can be found in a 2002 paper by Seth Gilbert and Nancy Lynch.[4]

A system that is not tolerant to network partitions can achieve data consistency and availability, and often does so by using transaction protocols. To make this work, client and storage systems must be part of the same environment; they fail as a whole under certain scenarios and as such clients cannot observe partitions. An important observation is that in larger distributed-scale systems, network partitions are a given; therefore, consistency and availability cannot be achieved at the same time. This means there are two choices on what to drop: relaxing consistency will allow the system to remain highly available under the partitionable conditions; making consistency a priority means that under certain conditions the system will not be available.

Both options require the client developer to be aware of what the system is offering. If the system emphasizes consistency, the developer has to deal with the fact that the system may not be available to take, for example, a write. If this write fails because of system unavailability, then the developer will have to deal with what to do with the data to be written. If the system emphasizes availability, it may always accept the write, but under certain conditions a read will not reflect the result of a recently completed write. The developer then has to decide whether the client requires access to the absolute latest update all the time. There is a range of applications that can handle slightly stale data, and they are served well under this model.

In principle the consistency property of transaction systems as defined in the ACID properties (atomicity,

ILLUSTRATIONS BY DAVE BOLLINGER

consistency, isolation, durability) is a different kind of consistency guarantee. In ACID, consistency relates to the guarantee that when a transaction is finished the database is in a consistent state; for example, when transferring money from one account to another the total amount held in both accounts should not change. In ACID-based systems, this kind of consistency is often the responsibility of the developer writing the transaction but can be assisted by the database managing integrity constraints.

Consistency—Client and Server

There are two ways of looking at consistency. One is from the developer/client point of view: how they observe data updates. The other is from the server side: how updates flow through the system and what guarantees systems can give with respect to updates.

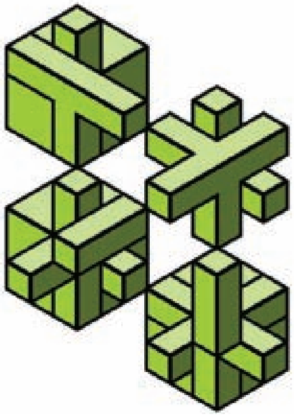
The components for the client side include:

▸ *A storage system.* For the moment we'll treat it as a black box, but one should assume that under the covers it is something of large scale and highly distributed, and that it is built to guarantee durability and availability.

▸ *Process A.* This is a process that writes to and reads from the storage system.

▸ *Process B and C.* These two processes are independent of process A and write to and read from the storage system. It is irrelevant whether these are really processes or threads within the same process; what is important is that they are independent and need to communicate to share information.

Client-side consistency has to do with how and when observers (in this case the processes A, B, or C) see updates made to a data object in the storage systems. In the following examples illustrating the different types of consistency, process A has made an update to a data object:



▸ *Strong consistency.* After the update completes, any subsequent access (by A, B, or C) will return the updated value.

▸ *Weak consistency.* The system does not guarantee that subsequent accesses will return the updated value. A number of conditions need to be met before the value will be returned. The period between the update and the moment when it is guaranteed that any observer will always see the updated value is dubbed the *inconsistency window*.

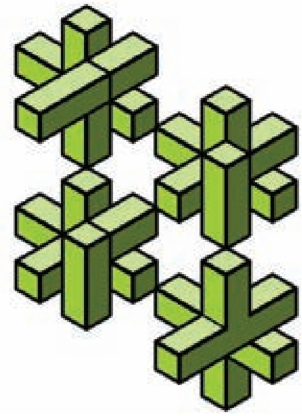
▸ *Eventual consistency.* This is a specific form of weak consistency; the storage system guarantees that if no new updates are made to the object, eventually all accesses will return the last updated value. If no failures occur, the maximum size of the inconsistency window can be determined based on factors such as communication delays, the load on the system, and the number of replicas involved in the replication scheme. The most popular system that implements eventual consistency is the domain name system (DNS). Updates to a name are distributed according to a configured pattern and in combination with time-controlled caches; eventually, all clients will see the update.

The eventual consistency model has a number of variations that are important to consider:

▸ *Causal consistency.* If process A has communicated to process B that it has updated a data item, a subsequent access by process B will return the updated value, and a write is guaranteed to supersede the earlier write. Access by process C that has no causal relationship to process A is subject to the normal eventual consistency rules.

▸ *Read-your-writes consistency.* This is an important model where process A, after having updated a data item, always accesses the updated value and never sees an older value. This is a special case of the causal consistency model.

▸ *Session consistency.* This is a practical version of the previous model, where a process accesses the storage system in the context of a session. As long as the session exists, the system guarantees read-your-writes consistency. If the session terminates because of a certain failure scenario, a new session must be created and the guarantees do not overlap the sessions.



▸ *Monotonic read consistency.* If a process has seen a particular value for the object, any subsequent accesses will never return any previous values.

▸ *Monotonic write consistency.* In this case, the system guarantees to serialize the writes by the same process. Systems that do not guarantee this level of consistency are notoriously difficult to program.

A number of these properties can be combined. For example, one can get monotonic reads combined with session-level consistency. From a practical point of view these two properties (monotonic reads and read-your-writes) are most desirable in an eventual consistency system, but not always required. These two properties make it simpler for developers to build applications, while allowing the storage system to relax consistency and provide high availability.

As you can see from these variations, quite a few different scenarios are possible. It depends on the particular applications whether or not one can deal with the consequences.

Eventual consistency is not some esoteric property of extreme distributed systems. Many modern RDBMSs (relational database management systems) that provide primary-backup reliability implement their replication techniques in both synchronous and asynchronous modes. In synchronous mode the replica update is part of the transaction. In asynchronous mode the updates arrive at the backup in a delayed manner, often through log shipping. In the latter mode if the primary fails before the logs are shipped, reading from the promoted backup will produce old, inconsistent values. Also to support better scalable read performance, RDBMSs have started to provide the ability to read from the backup, which is a classical case of providing eventual consistency guarantees in which the inconsistency windows depend on the periodicity of the log shipping.

On the server side we need to take a deeper look at how updates flow through the system to understand what drives the different modes that the developer who uses the system can experience. Let's establish a few definitions before getting started:

N = The number of nodes that store replicas of the data.

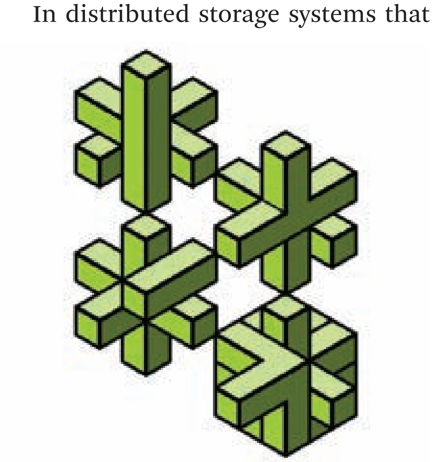
W = The number of replicas that need to acknowledge the receipt of the update before the update completes.

R = The number of replicas that are contacted when a data object is accessed through a read operation.

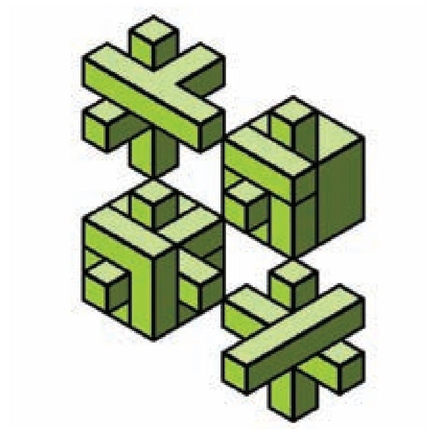
If W+R > N, then the write set and the read set always overlap and one can guarantee strong consistency. In the primary-backup RDBMS scenario, which implements synchronous replication, N=2, W=2, and R=1. No matter from which replica the client reads, it will always get a consistent answer. In the asynchronous replication case with reading from the backup enabled, N=2, W=1, and R=1. In this case R+W=N, and consistency cannot be guaranteed.

The problems with these configurations, which are basic quorum protocols, is that when because of failures the system cannot write to W nodes, the write operation has to fail, marking the unavailability of the system. With N=3 and W=3 and only two nodes available, the system will have to fail the write.

In distributed storage systems that



When a system processes trillions and trillions of requests, events that normally have a low probability of occurrence are now guaranteed to happen and must be accounted for upfront in the design and architecture of the system.



provide high performance and high availability the number of replicas is in general higher than two. Systems that focus solely on fault tolerance often use N=3 (with W=2 and R=2 configurations). Systems that must serve very high read loads often replicate their data beyond what is required for fault tolerance; N can be tens or even hundreds of nodes, with R configured to 1 such that a single read will return a result. Systems that are concerned with consistency are set to W=N for updates, which may decrease the probability of the write succeeding. A common configuration for these systems that are concerned about fault tolerance but not consistency is to run with W=1 to get minimal durability of the update and then rely on a lazy (epidemic) technique to update the other replicas.

How to configure N, W, and R depends on what the common case is and which performance path needs to be optimized. In R=1 and N=W we optimize for the read case, and in W=1 and R=N we optimize for a very fast write. Of course in the latter case, durability is not guaranteed in the presence of failures, and if W < (N+1)/2, there is the possibility of conflicting writes when the write sets do not overlap.

Weak/eventual consistency arises when W+R <= N, meaning that there is a possibility that the read and write set will not overlap. If this is a deliberate configuration and not based on a failure case, then it hardly makes sense to set R to anything but 1. This happens in two very common cases: the first is the massive replication for read scaling mentioned earlier; the second is where data access is more complicated. In a simple key-value model it is easy to compare versions to determine the latest value written to the system, but in systems that return sets of objects it is more difficult to determine what the correct latest set should be. In most of these systems where the write set is smaller than the replica set, a mechanism is in place that applies the updates in a lazy manner to the remaining nodes in the replica's set. The period until all replicas have been updated is the inconsistency window discussed before. If W+R <= N, then the system is vulnerable to reading from nodes that have not yet received the updates.

Whether or not read-your-write, session, and monotonic consistency can be achieved depends in general on the "stickiness" of clients to the server that executes the distributed protocol for them. If this is the same server every time, then it is relatively easy to guarantee read-your-writes and monotonic reads. This makes it slightly more difficult to manage load balancing and fault tolerance, but it is a simple solution. Using sessions, which are sticky, makes this explicit and provides an exposure level that clients can reason about.

Sometimes the client implements read-your-writes and monotonic reads. By adding versions on writes, the client discards reads of values with versions that precede the last-seen version.

Partitions happen when some nodes in the system cannot reach other nodes, but both sets are reachable by groups of clients. If you use a classical majority quorum approach, then the partition that has W nodes of the replica set can continue to take updates while the other partition becomes unavailable. The same is true for the read set. Given that these two sets overlap, by definition the minority set becomes unavailable. Partitions don't happen frequently, but they do occur between data centers, as well as inside data centers.

In some applications the unavailability of any of the partitions is unacceptable, and it is important that the clients that can reach that partition make progress. In that case both sides assign a new set of storage nodes to receive the data, and a merge operation is executed when the partition heals. For example, within Amazon the shopping cart uses such a write-always system; in the case of partition, a customer can continue to put items in the cart even if the original cart lives on the other partitions. The cart application assists the storage system with merging the carts once the partition has healed.

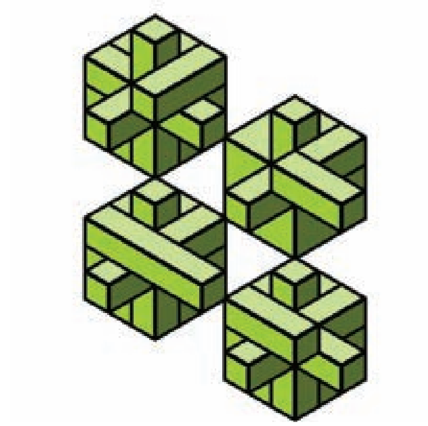
Amazon's Dynamo

A system that has brought all of these properties under explicit control of the application architecture is Amazon's Dynamo, a key-value storage system that is used internally in many services that make up the Amazon e-commerce platform, as well as Amazon's Web Services. One of the design goals of Dynamo is to allow the application service owner who creates an instance of the Dynamo storage system—which commonly spans multiple data centers—to make the trade-offs between consistency, durability, availability, and performance at a certain cost point.[3]

Summary

Data inconsistency in large-scale reliable distributed systems must be tolerated for two reasons: improving read and write performance under highly concurrent conditions; and handling partition cases where a majority model would render part of the system unavailable even though the nodes are up and running.

Whether or not inconsistencies are acceptable depends on the client application. In all cases the developer must be aware that consistency guarantees are provided by the storage systems and must be taken into account when developing applications. There are a number of practical improvements to the eventual consistency model, such as session-level consistency and monotonic reads, which provide better tools for the developer to work with. Many times the application is capable of handling the eventual consistency guarantees of the storage system without any



problem. A specific popular case is a Web site in which we can have the notion of user-perceived consistency. In this scenario the inconsistency window must be smaller than the time expected for the customer to return for the next page load. This allows for updates to propagate through the system before the next read is expected.

The goal of this article is to raise awareness about the complexity of engineering systems that need to operate at a global scale and that require careful tuning to ensure that they can deliver the durability, availability, and performance that their applications require. One of the tools the system designer has is the length of the consistency window, during which the clients of the systems are possibly exposed to the realities of large-scale systems engineering.

References

1. Brewer, E.A. Towards robust distributed systems (abstract). In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing* (July 16–19, 2000, Portland, OR), 7.
2. Conversation with Bruce Lindsay. *ACM Queue 2*, 8 (2004), 22–33.
3. DeCandia, G., et. al. Dynamo: Amazon's highly available key-value store. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles* (Stevenson, WA, Oct. 2007).
4. Gilbert, S. and Lynch, N. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant Web services. *ACM SIGACT News 33*, 2 (2002).
5. Lindsay, B.G. et al. Notes on distributed databases. *Distributed Data Bases.* I.W. Draffan and F. Poole, Eds. Cambridge University Press, Cambridge, MA, 1980, 247–284. Also available as IBM Research Report RJ2517, San Jose, CA (July 1979).

Werner Vogels is vice president and chief technology officer at Amazon.com, where he is responsible for driving the company's technology vision of continuously enhancing innovation on behalf of Amazon's customers at a global scale.

A previous version of this article appeared in the October 2008 issue of *ACM Queue*.

DOI:10.1145/1435417.1435431

Double, double toil and trouble
—Shakespeare, *Macbeth*, Act 4, Scene 1

BY JOHN MASHEY

The Long Road To 64 Bits

SHAKESPEARE'S WORDS OFTEN cover circumstances beyond his wildest dreams. *Toil and trouble* accompany major computing transitions, even when people plan ahead. Much of tomorrow's software will still be driven by decades-old decisions. Past decisions have unanticipated side effects that last decades and can be difficult to undo.

For example, consider the overly long, often awkward, and sometimes contentious process by which 32-bit microprocessor systems evolved into 64/32-bitters needed to address larger storage and run mixtures of 32- and 64-bit user programs. Most major general-purpose CPUs now have such versions, so bits have "doubled," but "toil and trouble" are not over, especially in software.

This example illustrates the interactions of hardware, languages (especially C), operating system, applications, standards, installed-base inertia, and industry politics. We can draw lessons ranging from high-level strategies down to programming specifics.

Fundamental Problem (late 1980s)

Running out of address space is a long tradition in computing, and often quite predictable. Moore's Law grew DRAM approximately four times bigger every three to four years, and by the mid-1990s, people were able to afford 2GB to 4GB of memory for midrange microprocessor systems, at which point simple 32-bit addressing (4GB) would get awkward. Ideally, 64/32-bit CPUs would have started shipping early enough (1992) to have made up the majority of the relevant installed base before they

Chronology: Multiple Interlocking Threads

IBM S/360
32-bit, with 24-bit addressing (16MB total) of real (core) memory

1964

1965

PHOTOGRAPH BY BEN FRANSKE

were actually needed. Then people could have switched to 64/32-bit operating systems and stopped upgrading 32-bit-only systems, allowing a smooth transition. Vendors naturally varied in their timing, but shipments ranged from "just barely in time" to "rather late." This is somewhat odd, considering the long, well-known histories of insufficient address bits, combined with the clear predictability of Moore's Law. All too often, customers were unable to use memory they could easily afford.

Some design decisions are easy to change, but others create long-term legacies. Among those illustrated here are:

- Some unfortunate decisions may be driven by real constraints (1970: PDP-11 16-bit).
- Reasonable-at-the-time decisions turn out in 20-year retrospect to have been suboptimal (1976–1977: usage of C data types). Some better usage recommendations could have saved a great deal of toil and trouble later.
- Some decisions yield short-term benefits but incur long-term problems (1964: S/360 24-bit addresses).
- Predictable trends are ignored, or transition efforts underestimated (1990s: 32 -> 64/32).

Constraints. Hardware people needed to build 64/32-bit CPUs at the right time—neither too early (extra cost, no market), nor too late (competition, angry customers). Existing 32-bit binaries needed to run on upward-compatible 64/32-bit systems, and they could be expected to coexist forever, because many would never need to be 64 bits. Hence, 32 bits could not be a temporary compatibility feature to be quickly discarded in later chips.

Software designers needed to agree on whole sets of standards; build dual-mode operating systems, compilers, and libraries; and modify application source code to work in both 32- and 64-bit environments. Numerous details had to be handled correctly to avoid redundant hardware efforts and maintain software sanity.

Solutions. Although not without subtle problems, the hardware was generally straightforward, and not that expensive—the first commercial 64-bit micro's 64-bit data path added at most 5% to the chip area, and this fraction dropped rapidly in later chips. Most chips used the same general approach of widening 32-bit registers to 64 bits. Software solutions were much more complex, involving arguments about 64/32-bit C, the nature of existing software, competition/cooperation among vendors, official standards, and influential but totally unofficial ad hoc groups.

Legacies. The IBM S/360 is 40 years old and still supports a 24-bit legacy addressing mode. The 64/32 solutions are at most 15 years old, but will be with us, effectively, forever. In 5,000 years, will some software maintainer still be muttering, "Why were they so dumb?"[4]

We managed to survive the Y2K problem—with a lot of work. We're still working through 64/32. Do we have any other problems like that? Are 64-bit CPUs enough to help the "Unix 2038" problem, or do we need to be working harder on that? Will we run out of 64-bit systems, and what will we do then? Will IPv6 be implemented widely enough soon enough?

All of these are examples of long-lived problems for which modest foresight may save later toil and trouble. But software is like politics: Sometimes we wait until a problem is really painful before we fix it.

Problem: CPU Must Address Available Memory

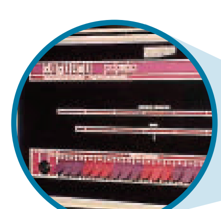
Any CPU can efficiently address some amount of virtual memory, done most conveniently by flat addressing, in which all or most of the bits in an integer register form a virtual memory address that may be more or less than actual physical memory. Whenever affordable physical memory exceeds the easily addressable, it stops being easy to throw memory at performance problems, and programming complexity rises quickly. Sometimes, segmented memory schemes have been used with varying degrees of success and programming pain. History is filled with awkward extensions that added a few bits to extend product life a few years, usually at the cost of hard work by operating-system people.

Moore's Law has increased affordable memory for decades. Disks have grown even more rapidly, especially since 1990. Larger disk pointers are more convenient than smaller ones, although less crucial than memory pointers. These interact when mapped files are used, rapidly consuming virtual address space.

In the mid-1980s, some people started thinking about 64-bit micros—for example, the experimental systems built by DEC (Digital Equipment Corporation). MIPS Computer Systems decided by late 1988 that its next design must be a true 64-bit CPU, and announced the R4000 in 1991. Many people thought MIPS was crazy or at least premature. I thought the system came just barely in time to develop software to match increasing DRAM, and I wrote an article to explain why.[5] The issues have not changed very much since then.

N-bit CPU. By long custom, an N-bit CPU implements an ISA (instruction

Algol 68
includes long long



DEC PDP-11/20
16-bit, 16-bit addressing (64KB total)

IBM S/370 family
virtual memory, 24-bit addresses, but multiple user address spaces allowed

1966 1967 **1968** 1969 **1970**

set architecture) with N-bit integer registers and N (or nearly N) address bits, ignoring sizes of buses or floating-point registers. Many 32-bit ISAs have 64- or 80-bit floating-point registers and implementations with 8-, 16-, 32-, 64-, or 128-bit buses. Sometimes marketers have gotten this confused. I use the term 64/32-bit here to differentiate the newer microprocessors from the older 64-bit word-oriented supercomputers, as the software issues are somewhat different. In the same sense, the Intel 80386 might have been called a 32/16-bit CPU, as it retained complete support for the earlier 16-bit model.

Why 2N-bits? People sometimes want wider-word computers to improve performance for parallel bit operations or data movement. If one needs a 2N-bit operation (add, multiply, and so on), each can be done in one instruction on a 2N-bit CPU, but requires longer sequences on an N-bit CPU. These are straightforward low-level performance issues. The typical compelling reason for wider words, however, has been the need to increase address bits, because code that is straightforward and efficient with enough address bits may need global restructuring to survive fewer bits.

Addressing—virtual and real—in a general-purpose system. User virtual addresses are mapped to real memory addresses, possibly with intervening page faults whereby the operating system maps the needed code or data from disk into memory. A user program can access at most VL (virtual limit) bytes, where VL starts at some hardware limit, then sometimes loses more space to an operating system. For example, 32-bit systems easily have VLs of 4, 3.75, 3.5, or 2GB. A given program execution uses at most PM (program memory) bytes of virtual memory. For many programs PM can differ greatly according to the input, but of course PM ≤ VL.

The RL (real limit) is visible to the operating system and is usually limited by the width of physical address buses. Sometimes mapping hardware is used to extend RL beyond a too-small "natural" limit (as happened in PDP-11s, described later). Installed AM (actual memory) is less visible to user programs and varies among machines without needing different versions of the user program.

Most commonly, VL ≥ RL ≥ AM. Some programs burn virtual address space for convenience and actually perform acceptably when PM >> AM: I've seen cases where 4:1 still worked, as a result of good locality. File mapping can increase that ratio further and still work. On the other hand, some programs run poorly whenever PM > AM, confirming the old proverb, "Virtual memory is a way to sell real memory."

Sometimes, a computer family starts with VL ≥ RL ≥ AM, and then AM grows, and perhaps RL is increased in ways visible only to the operating system, at which point VL << AM. A single program simply cannot use easily buyable memory, forcing work to be split and making it more difficult. For example, in Fortran, the declaration REAL X (M, M, M) is a three-dimensional array. If M = 100, *X* needs 4MB, but people would like the same code to run for M = 1,000 (4GB), or 6,300 (1,000GB). A few such systems do exist, although they are not cheap. I once had a customer complain about lack of current support for 1,000GB of real memory, although later the customer was able to buy such a system and use that memory in one program. After that, the customer complained about lack of 10,000GB support…

Of course, increasing AM in a multitasking system is still useful in improving the number of memory-resident tasks or reducing paging overhead, even if each task is still limited by VL. Operating system code is always simplified if it can directly and simply address all installed memory without having to manage extra memory maps, bank registers, and so on.

Running out of address bits has a long history.

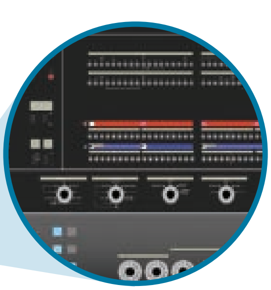
Mainframes, Minicomputers, Microprocessors

The oft-quoted George Santayana is apropos here: "Those who do not remember the past are condemned to repeat it."

Mainframes. IBM S/360 mainframes (circa 1964; see accompanying Chronology sidebar) had 32-bit registers, of which only 24 bits were used in addressing, for a limit of 16MB of core memory. This was considered immense at the time. A "large" mainframe offered at most 1MB of memory, although a few "huge" mainframes could provide 6MB. Most S/360s did not support virtual memory, so user programs generated physical addresses directly, and the installed AM was partitioned among the operating system and user programs. The 16MB limit was unfortunate, but ignoring (not trapping) the high-order 8 bits was worse. Assembly language programmers cleverly packed 8-bit flags with 24-bit addresses into 32-bit words.

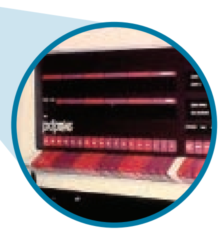
As virtual addressing S/370s (1970) enabled programs that were larger than physical memory allowed, and as core gave way to DRAM (1971), the 16MB limit grew inadequate. IBM 370XA CPUs (1983) added 31-bit addressing mode, but retained a (necessary) 24-bit mode for upward compatibility. I had

IBM 370/145 PHOTOGRAPH BY SELBST GEZEICHNET, DEC PDP-11/20 AND DEC PDP-11/20 PHOTOGRAPH BY JAY R. JAEGER



IBM 370/145
main memory no longer core, but DRAM, 1 Kbit/chip

DEC PDP-11/45
separate instruction+data (64KI + 64KD); 248KB maximum real memory



Unix
PDP-11/45, operating system rewritten in C; IP16

C
integer data types: int, char; C on other machines (36-bit Honeywell 6000, IBM 370, others)

Unix
sixth edition: 24-bit maximum file size (16MB)

1971 1972 **1973** 1974 **1975**

been one of those "clever" programmers and was somewhat surprised to discover that a large program (the S/360 ASSIST assembler) originally written in 1970 was still in use in 2006—in 24-bit compatibility mode, because it wouldn't run any other way. Compiler code that had been "clever" had long since stopped doing this, but assembly code is tougher. ("The evil that men do lives after them, the good is oft interred with their bones." —Shakespeare, again, *Julius Caesar*)

Then, even 31-bit addressing became insufficient for certain applications, especially databases. ESA/370 (1988) offered user-level segmentation to access multiple 2GB regions of memory.

The 64-bit IBM zSeries (2001) still supports 24-bit mode, 40-plus years later. Why did 24-bit happen? I'm told that it was all for the sake of one low-cost early model, the 360/30, where 32 bits would have run slower because it had 8-bit data paths. These were last shipped more than 30 years ago. Were they worth the decades of headaches?

Minicomputers. In the 16-bit DEC PDP-11 minicomputer family (1970), a single task addressed only 64KB, or in later models (1973), 64KB of instructions plus 64KB of data. "The biggest and most common mistake that can be made in computer design is that of not providing enough address bits for memory addressing and management," C. Gordon Bell and J. Craig Mudge wrote in 1978. "The PDP-11 followed this hallowed tradition of skimping on address bits, but was saved on the principle that a good design can evolve through at least one major change. For the PDP-11, the limited address space was solved for the short run, but not with enough finesse to support a large family of minicomputers. This was indeed a costly oversight."[2]

To be fair, it would have been difficult to meet the PDP-11's cost goals with 32-bit hardware, but I think DEC did underestimate how fast the price of DRAM memory would fall. In any case, this lasted a long time—the PDP-11 was finally discontinued in 1997!

The PDP-11/70 (1976) raised the number of supportable concurrent tasks, but any single program could still only use 64KI + 64KD of a maximum of 4MB, so that individual large programs required unnatural acts to split code and data into 64KB pieces. Some believed this encouraged modularity and inhibited "creeping featurism" and was therefore philosophically good.

Although the 32-bit VAX-11/780 (1977) was only moderately faster than the PDP-11/70, the increased address space was a major improvement that ended the evolution of high-end PDP-11s. VAX architect William Strecker explained it this way: "However, there are some applications whose programming is impractical in a 65KB address space, and perhaps more importantly, others whose programming is appreciably simplified by having a large address space."[7]

Microprocessors. The Intel 8086's 16-bit ISA seemed likely to fall prey to the PDP-11's issues (1978). It did provide user-mode mechanisms for explicit segment manipulation, however. This allowed a single program to access more than 64KB of data. PC programmers were familiar with the multiplicity of memory models, libraries, compiler flags, extenders, and other artifacts once needed. The 80386 provided 32-bit flat addresses (1986), but of course retained the earlier mechanisms, and 16-bit PC software lasted "forever." The intermediate 80286 (1982) illustrated the difficulty of patching an architecture to get more addressing bits.

The 32-bit Motorola MC68000 (1979) started with a flat-addressing programming model. By ignoring the high 8 bits of a 32-bit register, it exactly repeated the S/360 mistake. Once again, "clever" programmers found uses for those bits, and when the MC68020 (1984) interpreted all 32, some programs broke (for example, when moving from the original Apple Macintosh to the Mac II).

Fortunately, 64-bit CPUs managed to avoid repeating the S/360 and MC68000 problem. Although early versions usually implemented 40 to 44 virtual address bits, they trapped use of not-yet-implemented high-order v bits, rather than ignoring them. People do learn, eventually.

Lessons

▸ Even in successful computer families created by top architects, address bits are scarce and are totally consumed sooner or later.

▸ Upward compatibility is a real constraint, and thinking ahead helps. In the mainframe case, a 24-bit "first-implementation artifact" needed hardware/software support for 40-plus years. Then a successful minicomputer family's evolution ended prematurely. Finally, microprocessors repeated the earlier mistakes, although the X86's segmentation allowed survival long enough to get flat-address versions.

DEC PDP-11/70
(64KI + 64KD), but larger physical memory (a huge 4MB)

C
short, long added (partly from doing C for XDS Sigma, although long was 64 bits there)

1976

Unix
ported to 32-bit Interdata 8/32

C
unsigned, typedef, union; 32-bit long used to replace int,2 in lseek, tell on 16-bit PDP-11; IP16L32

DEC VAX-11/780
32-bit, 32-bit addressing (4GB total, 2GB per user process)

C
PDP-11: I16LP32; VAX (other 32-bitters): ILP32

1977

Unix
32V for VAX-11/780; C is ILP32

C
The C Programming Language, Brian Kernighan and Dennis Ritchie (Prentice-Hall)

Intel 8086
16-bit, but with user-visible segmentation

1978

Motorola MC68000
32-bit ISA, but 24-bit addressing (e.g., S/360)

1979

1980

The 32- to 64-bit Problem in the Late 1980s

By the late 1980s, Moore's Law seemed cast in silicon, and it was clear that by 1993–1994, midrange microprocessor servers could cost-effectively offer 2GB–4GB or more of physical memory. We had seen real programs effectively use as much as 4:1 more virtual memory than installed physical memory, which meant pressure in 1993–1994, and real trouble by 1995. As I wrote in *Byte* in September 1991:[5]

"The virtual addressing scheme often can exceed the limits of possible physical addresses. A 64-bit address can handle literally a mountain of memory: Assuming that 1 megabyte of RAM requires 1 cubic inch of space (using 4-megabit DRAM chips), 2**64 bytes would require a square mile of DRAM piled more than 300 feet high! For now, no one expects to address this much DRAM, even with next-generation 16MB DRAM chips, but increasing physical memory slightly beyond 32 bits is definitely a goal. With 16MB DRAM chips, 2**32 bytes fits into just over 1 cubic foot (not including cooling)—feasible for deskside systems…

"Database systems often spread a single file across several disks. Current SCSI disks hold up to 2 gigabytes (i.e., they use 31-bit addresses). Calculating file locations as virtual memory addresses requires integer arithmetic. Operating systems are accustomed to working around such problems, but it becomes unpleasant to make workarounds; rather than just making things work well, programmers are struggling just to make something work…."

So, people started to do something about the problem.

SGI (Silicon Graphics). Starting in early 1992, all new SGI products used only 64/32-bit chips, but at first they still ran a 32-bit operating system. In late 1994, a 64/32-bit operating system and compilers were introduced for large servers, able to support both 32-bit and 64-bit user programs. This software worked its way down the product line. A few customers quickly bought more than 4GB of memory and within a day had recompiled programs to use it, in some cases merely by changing one Fortran parameter. Low-end SGI workstations, however, continued to ship with a 32-bit-only operating system for years, and of course, existing 32-bit hardware had to be supported…for years. For historical reasons, SGI had more flavors of 32-bit and 64-bit instruction sets than were really desirable, so it was worse than having just two of them.

This is the bad kind of "long tail"—people focus on "first ship dates," but often the "last ship date" matters more, as does the "last date on which we will release a new version of an operating system or application that can run on that system." Windows 16-bit applications still run on regular Windows XP, 20 years after the 80386 was introduced. Such support has finally been dropped in Windows XP x64.

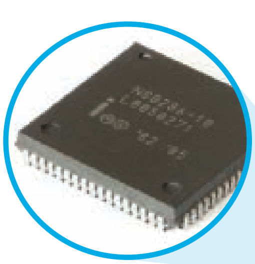
DEC. DEC shipped 64-bit Alpha systems in late 1992, with a 64-bit operating system, and by late 1994 was shipping servers with memories large enough to need greater than 32-bit addressing. DEC might have requested (easy) 32-bit ports, but thinking long term, it went straight to 64-bit, avoiding duplication. It was expensive in time and money to get third-party software 64-bit clean, but it was valuable to the industry as it accelerated the 64-bit cleanup. DEC was probably right to do this, since it had no installed base of 32-bit Alpha programs and could avoid having to support two modes. For VMS, early versions were 32-bit, and later ones 64/32-bit.

Other vendors. Over the next few years, many vendors shipped 64-bit CPUs, usually running 32-bit software, and later 64/32-bit: Sun UltraSPARC (1995), HAL SPARC64 (1995), PA-RISC (1996), HP/UX 11.0 (1997), IBM RS64 and AIX 4.3 (1997), Sun Solaris 7 (1998), IBM zSeries (2001), Intel Itanium (2001), AMD AMD64 (2003), Intel EM-T64a (2004), Microsoft Windows XP x64 (2005). Linux 64-bit versions appeared at various times.

Most 64-bit CPUs were designed as extensions of existing 32-bit architectures that could run existing 32-bit binaries well, usually by extending 32-bit registers to 64 bits in 64-bit mode, but ignoring the extra bits in 32-bit mode. The long time span for these releases arises from natural differences in priorities. SGI was especially interested in high-performance technical computing, whose users were accustomed to 64-bit supercomputers and could often use 64 bits simply by increasing one array dimension in a Fortran program and recompiling. SGI and other vendors of large servers also cared about memory for large database applications. It was certainly less important to X86 CPU vendors whose volume was dominated by PCs. In Intel's case, perhaps the emphasis on Itanium delayed 64-bit X86s.

By 2006, 4GB of DRAM, typically consisting of 1GB DRAMs, typically used

INTEL 80286 PHOTOGRAPH BY HENRY MÜHLPFORDT, CONVEX C4600 KEVIN D. KISSELL



C
I16LP32 on MC68000
in Bell Labs Blit terminal

Intel 80286
allows 16MB of real memory, but restrictions keep most systems at 1MB

IBM 370/XA
adds 31-bit mode for user programs, 24-bit mode still supported

C
Unix workstations generally use ILP32, following Unix on VAX systems

Motorola MC68020
32-bit; 32-bit addressing

C
Amdahl UTS (32-bit S/370) uses long long, especially for large file pointers

C
Convex (64-bit vector mini-supercomputer) uses long long for 64-bit integers

1981 | 1982 | 1983 | 1984 | 1985

Common C data types.

int	long	ptr	long long	Label	Examples
16		16		IP16	PDP-11 Unix (early, 1973)
16	32	16		IP16L32	PDP-11 Unix (later, 1977) Multiple instructions for long
16	32	32		I16LP32	Early MC68000 (1982) Apple Macintosh 68 KB Microsoft operating systems (plus extras for X86 segments)
32	32	32		ILP32	IBM 370; VAX Unix; many workstations
32	32	32	64	ILP32LL or	Amdahl; Convex; 1990s Unix systems
				ILP32LL64	Like IP15L32, for same reason; multiple instructions for long long.
32	32	32	64	LLP64 or IL32LLP64 or P64	Microsoft Win64
32	32	32	*(64)	LP64 or I32LP64	Most 64/32 Unix systems
64	32	32	*(64)	ILP64	HAL; logical analog of ILP32

* In these cases, LP64 and ILP64 offer 64-bit integers, and long long seems redundant, but in practice, most proponents of LP64 and ILP64 included long long as well, for reasons given later. ILP64 uniquely required a new 32-bit type, usually called _int32.

four DIMMs and could cost less than $400, 300GB disks are widely available for less than $1 per GB, so one would have expected mature, widespread support for 64 bits by then. All this took longer than perhaps it should have, however, and there have been many years where people could buy memory but not be able to address it conveniently, or not be able to buy some third-party application that did, because such applications naturally lag 64-bit CPUs. It is worth understanding why and how this happened, even though the impending issue was well known.

Lessons

▸ For any successful computer family, it takes a very long time to convert an installed base of hardware, and software lasts even longer.

▸ Moving from 32 to 64/32 bits is a long-term coexistence scenario. Unlike past transitions, almost all 64-bit CPUs must run compatible existing 32-bit binaries without translation, since much of the installed base remains 32-bit and a significant number of 32-bit programs are perfectly adequate and can remain so indefinitely.

Software is Harder

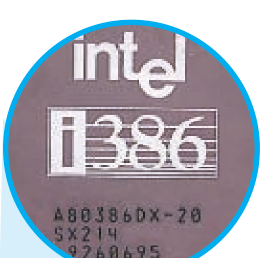
Building a 64-bit CPU is not enough. Embedded-systems markets can move easier than general-purpose markets, as happened, for example, with Cisco routers and Nintendo N64s that used 64-bit MIPS chips. Vendors of most 32-bit systems, however, had to make their way through all of the following steps to produce useful upward-compatible 64-bit systems:

1. Ship systems with 64/32 CPUs, probably running in 32-bit mode. Continue supporting 32-bit-only CPUs as long as they are shipped and for years thereafter (often five or more years). Most vendors did this, simply because software takes time.

2. Choose a 64-bit programming model for C, C++, and other languages. This involves discussion with standards bodies and consultation with competitors. There may be serious consequences if you select a different model from most of your competitors. Unix vendors and Microsoft did choose differently, for plausible reasons. Think hard about inter-language issues—Fortran expects INTEGER and REAL to be the same size, which makes 64-bit default integers awkward.

3. Clean up header files, carefully.

4. Build compilers to generate 64-bit code. The compilers themselves almost certainly run in 32-bit mode and cross-compile to 64-bit, although occasional huge machine-generated programs can demand compilers that run in 64-bit mode. Note that programmer sanity normally requires a bootstrap step here, in which the 32-bit compiler is first modified to accept 64-bit integers and then is recoded to use them itself.



Intel
80386, 32-bit, with support for 8086 mode

Apple Mac II
MC68020's 32-bit addressing causes trouble for some MC68000 software

IBM ESA/370
multiple 31-bit address spaces per user, although complex; 24-bit still there

ANSI C ("C89")
effort had started in 1983, ANSI X3J11

1986 **1987** **1988** **1989** **1990**

5. Convert the operating system to 64-bit, but with 32-bit interfaces as well, to run both 64- and 32-bit applications.

6. Create 64-bit versions of all system libraries.

7. Ship the new operating system and compilers on new 64-bit hardware, and hopefully, on the earlier 64-bit hardware that has now been shipping for a while. This includes supporting (at least) two flavors of every library.

8. Talk third-party software vendors into supporting a 64-bit version of any program for which this is relevant. Early in such a process, the installed base inevitably remains mostly 32-bit, and software vendors consider the potential market size versus the cost of supporting two versions on the same platform. DEC helped the industry fix 32- to 64-bit portability issues by paying for numerous 64-bit Alpha ports.

9. Stop shipping 32-bit systems (but continue to support them for many years).

10. Stop supporting 32-bit hardware with new operating system releases, finally.

11. Going from step 1 to step 6 typically took two to three years, and getting to step 9 took several more years. The industry has not yet completed step 10.

Operating system vendors can avoid step 1, but otherwise, the issues are similar. Many programs need never be converted to 64-bit, especially since many operating systems already support 64-bit file pointers for 32-bit programs.

Next, I trace some of the twists and turns that occurred in the 1990s, especially involving the implementation of C on 64/32-bit CPUs. This topic generated endless and sometimes vituperative discussions.

C: 64-bit Integers on 64/32-bit CPUs: Technology and Politics

People have used various (and not always consistent) notations to describe choices of C data types. In the accompanying table, the first label of several was the most common, as far as I can tell. On machines with 8-bit `char`, `short` is usually 16 bits, but other data types can vary. The common choices are shown in the table.

Early days. Early C integers (1973) included only `int` and `char`; then `long` and `short` were added by 1976, followed by `unsigned` and `typedef` in 1977. In the late 1970s, the installed base of 16-bit PDP-11s was joined by newer 32-bit systems, requiring that source code be efficient, sharable, and compatible between 16-bit systems (using I16LP32) and 32-bitters (ILP32), a pairing that worked well. PDP-11s still employed (efficient) 16-bit `int` most of the time, but could use 32-bit `long` as needed. The 32-bitters used 32-bit `int` most of the time, which was more efficient, but could express 16-bit via `short`. Data structures used to communicate among machines avoided `int`. It was very important that 32-bit `long` be usable on the PDP-11. Before that, the PDP-11 needed explicit functions to manipulate `int[2]` (16-bit `int` pairs), and such code was not only awkward, but also not simply sharable with 32-bit systems. This is an extremely important point—`long` was not strictly necessary for 32-bit CPUs, but it was very important to enable code sharing among 16- and 32-bit environments. We could have gotten by with `char`, `short`, and `int`, if all our systems had been 32 bits.

It is important to remember the nature of C at this point. It took a while for `typedef` to become common idiom. With 20/20 hindsight, it might have been wise to have provided a standard set of `typedefs` to express "fast integer," "guaranteed to be exactly N-bit integer," "integer large enough to hold a pointer," and to have recommended that people build their own `typedefs` on these definitions, rather than base types. If this had been done, perhaps much toil and trouble could have been avoided.

This would have been very countercultural, however, and it would have required astonishing precognition. Bell Labs already ran C on 36-bit CPUs and was working hard on portability, so overly specific constructs such as "int16" would have been viewed with disfavor. C compilers still had to run on 64KI+64KD PDP-11s, so language minimality was prized. The C/Unix community was relatively small (600 systems) and was just starting to adapt to the coming 32-bit minicomputers. In late 1977, the largest known Unix installation had seven PDP-11s, with a grand total of 3.3MB of memory and 1.9GB of disk space. No one could have guessed how pervasive C and its offshoots would become, and thinking about 64-bit CPUs was not high on the list of issues.

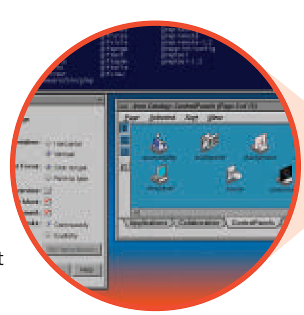
32-bit happy times. In the 1980s, ILP32 became the norm, at least in Unix-based systems. These were happy times: 32-bit was comfortable enough for buyable DRAM, for many years. In retrospect, however, it may have caused some people to get sloppy in assuming

APPLE MAC II PHOTOGRAPH BY ALEXANDER SCHAELSS. MIPS R4000 PHOTOGRAPH BY DIRK OPPELT

SGI
ships first 64-bit micro (MIPS R4000) in system; still running 32-bit operating system

Informal 64-bit C working group
discusses various models (LP64, ILP64, LLP64), with no agreement except to use long long and have inttypes.h to help users

DEC
ships 64-bit Alpha systems, running 64-bit operating system; LP64



SGI
ships IRIX 6 (64/32 operating system; ILP32LL + LP64) on Power Challenge; customers buy 4GB+ memory, use it

DEC
ships 4GB+ in DEC 7000 SMPs (may have been slightly earlier)

Sun UltraSPARC
64/32-bit hardware, 32-bit-only operating system

HAL Computer's SPARC64
uses ILP64 model for C

Large file summit
codifies 64-bit interface to files >2GB, even in 32-bit systems (ILP32LL+LP64)

Aspen group
supports LP64 model for C, so that Unix vendors are consistent

1991 | 1992 | 1993 | 1994 | 1995

`sizeof(int) == sizeof(long) == sizeof(ptr) == 32.`

Sometime around 1984, Amdahl UTS and Convex added `long long` for 64-bit integers, the former on a 32-bit architecture, the latter on a 64-bitter. UTS used this especially for long file pointers, one of the same motivations for long in PDP-11 Unix (1977). Algol 68 inspired long long in 1968, and it was also added to GNU C at some point. Many reviled this syntax, but at least it consumed no more reserved keywords.

Of course, 16-bit `int` was used on Microsoft DOS and Apple Macintosh systems, given the original use of Intel 8086 or MC68000, where 32-bit `int` would have been costly, particularly on early systems with 8- or 16-bit data paths and where low memory cost was especially important.

64-bit heats up in 1991/1992. The MIPS R4000 and DEC Alpha were announced in the early 1990s. Email discussions were rampant among various companies during 1990–1992 regarding the proper model for 64-bit C, especially when implemented on systems that would still run 32-bit applications for many years. Quite often, such informal cooperation exists among engineers working for otherwise fierce competitors.

In mid-1992 Steve Chessin of Sun initiated an informal 64-bit C working group to see if vendors could avoid implementing randomly different 64-bit C data models and nomenclatures. Systems and operating system vendors all feared the wrath of customers and third-party software vendors otherwise. DEC had chosen LP64 and was already far along, as Alpha had no 32-bit version. SGI was shipping 64-bit hardware and working on 64-bit compilers and operating system; it preferred LP64 as well. Many others were planning 64-bit CPUs or operating systems and doing portability studies.

Chessin's working group had no formal status, but had well-respected senior technologists from many systems and operating systems vendors, including several who were members of the C Standards Committee. With all this brainpower, one might hope that one clear answer would emerge, but that was not to be. Each of the three proposals (LLP64, LP64, and ILP64) broke different kinds of code, based on the particular implicit assumptions made in the 32-bit happy times.

Respected members of the group made credible presentations citing massive analyses of code and porting experience, each concluding, "*XXX* is the answer." Unfortunately, *XXX* was different in each case, and the group remained split three ways. At that point I suggested we perhaps could agree on header files that would help programmers survive (leading to `<inttypes.h>`). Most people did agree that `long long` was the least bad of the alternative notations and had some previous usage.

We worried that we were years ahead of the forthcoming C standard, but could not wait for it, and the C Standards Committee members were supportive. If we agreed on anything reasonable, and it became widespread practice, it would at least receive due consideration. Like it or not, at that point this unofficial group probably made `long long` inevitable—or perhaps that inevitability dated from 1984.

By 1994, DEC was shipping large systems and had paid for many third-party software ports, using LP64. SGI was also shipping large systems, which supported both ILP32LL and LP64, with `long long` filling the role handled by long in the late 1970s.

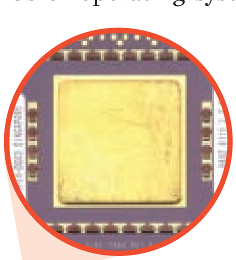
The DEC effort proved that it was feasible to make much software 64-bit-clean without making it 32-bit unclean. The SGI effort proved that 32-bit and 64-bit programs could be sensibly supported on one system, with reasonable data interchange, a requirement for most other vendors. In practice, that meant that one should avoid `long` in structures used to interchange data, exactly akin to the avoidance of `int` in the PDP-11/VAX days. About this time in 1995 the Large File Summit agreed on Unix APIs to increase file size above 2GB, using `long long` as a base data type.

Finally, the Aspen Group in 1995 had another round of discussions about the 64-bit C model for Unix and settled on LP64, at least in part because it had been proved to work and most actual 64-bit software used LP64.

During the 1992 meetings of the 64-bit C group Microsoft had not yet chosen its model, but later chose LLP64, not the LP64 preferred by Unix vendors. I was told that this happened because the only 32-bit integer type in PCs was `long`; hence, people often used `long` to mean 32-bit more explicitly than in Unix code. That meant that changing its size would tend to break more code than in Unix. This seemed plausible to me. Every choice broke some codes, and thus people looked at their own code bases, which differed, leading to reasonable differences of opinion.

Many people despised `long long` and filled newsgroups with arguments against it, even after it became incorporated into the next C standard in 1999. The official rationale for the C standard can be consulted by anyone who wants to understand the gory details.[6]

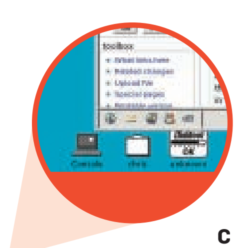
Differing implicit assumptions about sizes of various data types had grown up over the years and caused a great deal of confusion. If we could go



HP
announces PA-RISC 2.0, 64-bit

HP
UP/UX 11.0 is 64/32-bit OS; ILP32LL + LP64

IBM
RS64 PowerPC, AIX 4.3; ILP32LL + LP64



Sun
64/32 Solaris 7 released; ILP32LL + LP64

C
ISO/IEC C (WG14's "C99"); includes long long, at least 64 bits

1996 1997 1998 1999 2000

back to 1977, knowing what we know now, we could have made all this easier simply by insisting on more consistent use of `typedef`. In 1977, however, it would have been difficult to think ahead 20 years to 64-bit micros—we were just getting to 32-bit minis!

This process might also have been eased if more vendors had been further along in 64-bit implementations in 1992. Many people had either forgotten the lessons of the PDP-11/VAX era or had not been involved then. In any case, the 64/32 systems had a more stringent requirement: 64-bit and 32-bit programs would coexist forever in the same systems.

Timing is important to creating good industry standards. Too early, standards may be written with insufficient practical experience because no one has real implementations from which to learn. Too late, and vendors may well have committed to numerous incompatible implementations. In between, a few groups have early implementations to help inform the standards. Perhaps by accident, the 64-bit C standards were reasonably well-timed.

Lessons

▸ Standards often get created in nonstandard ways, and often, de facto standards long precede official ones.

▸ Reasonable people can disagree, especially when looking at different sets of data.

▸ Sometimes one must work with competitors to make anything reasonable happen.

▸ Programmers take advantage of extra bits or ambiguity of specification. Most of the arguments happen because application programmers make differing implicit assumptions.

▸ Code can be recompiled, but once data gets written somewhere, any new code must still be able to describe it cleanly. Current software is rarely done from scratch but has to exist inside a large ecosystem.

▸ We might never build 128-bit computers, but it would probably be good to invent a notation for 128-bit integers, whose generated code on 64-bit CPUs is about the same as 64-bit code is on 32-bit CPUs. It would be nice to do that long before it is really needed. In general, predictable long-term problems are most efficiently solved with a little planning, not with frenzied efforts when the problem is imminent. Fortunately, 128-bitters are many years away, if ever (maybe 2020–2040), because we've just multiplied our addressing size by four billion, and that will last a while, even if Moore's Law continues that long! In case 128-bit happens in 2020, however, it would be wise to be thinking about the next integer size around 2010.

Of course, when the S/360 was introduced, IBM and other vendors had 36- and 18-bit product lines. In an alternate universe, had the S/360 been a 36-bit architecture with four 9-bit bytes/word, most later machines would have been 18- and 36-bit, and we would just be starting the 36-bit to 72-bit transition.

▸ Hardware decisions last a long time, but software decisions may well last longer. If you are a practicing programmer, take pity on those who end up maintaining your code, and spend some time thinking ahead. Allow for natural expansion of hardware, hide low-level details, and use the highest-level languages you can. C's ability to make efficient use of hardware can be both a blessing and a curse.

Conclusion

Some decisions last a very long time. The 24-bit addressing of 1964's S/360 is still with us, as are some side effects of C usage in the mid-1970s. The transition to 64-bit probably took longer than it needed for a host of reasons. It's too bad that people quite often have been unable to use affordable memory for solving performance problems or avoiding cumbersome programming.

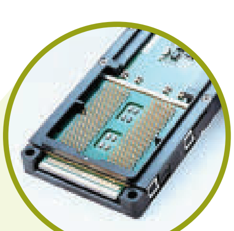
It's too bad there has been so much "toil and trouble," but "double" for microprocessors has been accomplished, and "double" for software is at least under way, and people have stopped arguing about the need for 64-bit micros.

References

1. Aspen Data Model Committee. 64-bit programming models: Why LP64? (1997–1998); www.unix.org/version2/whatsnew/lp64_wp.html.
2. Bell, C.G. and Mudge, J.C. The evolution of the PDP-11. *Computer Engineering: A DEC View of Computer System Design*. C.G. Bell, J.C. Mudge, and J.E. McNamara, Eds. Digital Press, Bedford, MA, 1978.
3. Josey, A. Data size neutrality and 64-bit support (1997); www.usenix.org/publications/login/standards/10.data.html.
4. Mashey, J. Languages, levels, libraries, longevity. *ACM Queue* 2, 9 (2004–2005), 32–38.
5. Mashey, J. 64-bit computing. *BYTE* (Sept. 1991), 135–142. 9 (The complete text can also be found by searching Google Groups comp.arch: Mashey BYTE 1991).
6. Rationale for International Standard—Programming Languages—C; www.open-std.org/jtc1/sc22/wg14/www/docs/n897.pdf (or other sites).
7. Strecker, W.D. VAX-11/780: A virtual address extension to the DEC PDP-11 family. *Computer Engineering: A DEC View of Computer System Design*. C.G. Bell, J.C. Mudge, and J.E. McNamara, Eds. Digital Press, Bedford, MA, 1978.
8. Unix specification; adding support for arbitrary file sizes; www.unix.org/version2/whatsnew/lfs20mar.html.

John Mashey is a consultant for venture capitalists and technology companies, sits on various technology advisory boards of startup companies, and is a trustee of the Computer History Museum. Along with spending 10 years at Bell Laboratories working on the Programmer's Workbench version of Unix, he was also one of the designers of the MIPS RISC architecture, one of the founders of the SPEC benchmarking group, and chief scientist at Silicon Graphics.

A previous version of this article appeared in the October 2006 issue of *ACM Queue*.



IBM
64-bit zSeries (S/360 descendent); 24-bit addressing still supported

Intel
64-bit Itanium

2001

Microsoft
Windows 64-bit for Itanium

2002

AMD
64-bit X86 (now called AMD64)

2003

Intel
64-bit X86, (called EMT64), compatible with AMD

2004

Microsoft
Windows XP Professional x64 for X86; LLP64 (or IL32LLP64)

2005

DOI:10.1145/1435417.1435433

Information systems enable rural development by increasing the accountability of nongovernmental organizations.

BY TAPAN S. PARIKH

Engineering Rural Development

ECONOMISTS AND ECOLOGISTS agree that the foremost challenge facing the world today is how to raise international standards of living while reducing humanity's environmental footprint. Also generally recognized is that the response must be global, including all human societies, cultures, and vocations. Nowhere is this more clearly articulated than in the Millennium Development Goals (www.un.org/millenniumgoals/), a mission statement for the world endorsed in 2000 by the 192 member states of the United Nations.

Access to information and communications technologies (ICTs) plays an increasingly important role in creating efficient markets and sustainable economic relationships. This is true not only for businesses but for initiatives in conservation, poverty reduction, and global health as well. To succeed, these efforts must be able to manage, understand, and react to globally distributed sources of information. Since the 1950s, advances in computing and other ICTs have revolutionized the way we create and share information, spurring an empirical revolution in fields as diverse and important as medicine, astronomy, biochemistry, economics, and ecology.

Understanding and responding to the global challenge of development must draw upon the same empirical and scientific methods. Failure to do so dooms us to repeat the same mistakes over and over. Computing technologies are essential for understanding the problems we face, as well as for planning, executing, and evaluating potential solutions. However, this process cannot be limited to the technical, social, and economic elite. To address historical disparities, it is essential that the poorest and least developed communities also contribute to, and benefit from, the global pool of knowledge.

Many rural ICT projects in developing countries have focused on individuals as the direct beneficiaries of technology. This includes efforts to improve rural education (such as the One Laptop Per Child project, laptop.org) and broader efforts to enable rural service delivery (such as the global movement to establish PC-based telecenters, or kiosks, www.telecentre.org). The early days of computing in the developed world emphasized business automation. Similarly, in rural areas of the developing world, one major initial benefit of ICTs will be the improvement, performance, and transparency of local institutions.

Nongovernmental organizations and community-based organizations, or NGOs and CBOs, can serve as an institutional basis for ICT adoption and use in the rural developing world. Due to their local knowledge and relationships, they are a key source of innovation, envisioning, implementing, and validating new models for rural

One farm inspector teaches another how to use Digital ICS in a village outside of Barillas, Guatemala, during an early field visit, spring 2006. (Photo by Ronak Parikh, now a student at Columbia University, then an intern at the University of Washington.)

development. Essentially, this is engineering, the process of generating, evaluating, and generalizing solutions to problems. Computing can play a critical role, enabling new and innovative solutions and better ways to demonstrate their contribution to human well being.

I start here with an overview of NGOs and CBOs and the value chain for recognizing and rewarding their work. To illustrate, I describe two information systems my students and I developed for improving NGO efficiency and accountability. I conclude by further describing the inherent engineering process within rural development and how computing can help enable it.

Civil Society

"Men are not birds," wrote Garrett Hardin, a professor of biology at the University of California, Santa Barbara, implying that for millennia humanity has abandoned the path of the animals (such as birds that would peck siblings to death when faced with the prospect of overpopulation[8]). The only civilized solution to the "tragedy of the commons," wrote Hardin, was "mutual coercion, mutually agreed upon by the majority of the people affected." Mutual coercion, or cooperation, is the basis for order in all human societies.

Civil society is an umbrella term for many kinds of voluntary and cooperative activities. Civil society organizations are nonprofit, non-state organizations representing people's social, economic, political, cultural, and environmental interests. In the developed world, they include charities, foundations, advocacy groups, and a variety of social service organizations. In the developing world, they are usually called NGOs, which, according to the World Bank, are characterized by their "voluntary and altruistic approach."[22]

NGOs focus on any number of issues, including educating children, empowering women, stemming rural migration, conserving the environment, and linking farmers to markets. They are characterized as operational or advocacy, depending on whether they "work in the field" designing and implementing specific development projects or influence policy and public opinion. Here, I use the term NGO to refer primarily to the operational NGOs working for socioeconomic development in rural areas of the developing world, including Latin America, South Asia, and Sub-Saharan Africa.

Where state and business infrastructure is limited or inaccessible, NGOs play a key role in organizing and supporting isolated communities, helping them access services like health care, education, training, and business development. By raising the standard of living and exposing people to new opportunities, NGOs are catalysts for change in rural areas throughout the developing world.

The World Bank classifies development NGOs into three categories: international, national, and commu-

Parikh explains to NGO field staff member how to use the Self-Help MIS mobile data-collection system while conducting usability studies in a village outside Madurai, Tamil Nadu, India, summer 2005. (University of California, Berkeley.)

nity-based, or CBOs.[22] International and national NGOs work through local offices, sometimes collaborating with smaller local NGOs and CBOs, to design, execute, and monitor development projects.

CBOs (also referred to as grassroots organizations) are distinct in that they are often membership-based and so directly represent a particular local community. Some CBOs also provide business services. For example, agricultural cooperatives allow small-holder farmers to achieve economies of scale in production, processing, marketing, and sales. Financial cooperatives (and smaller, informal entities, like mutual-liability microfinance groups) aggregate capital, conduct transactions, and distribute monitoring costs to enable convenient, affordable access to financial services. Larger NGOs are often catalysts in the formation, training, and capacity-building of CBOs.

NGOs function as change agents in rural areas by identifying promising opportunities for socio-economic development and then supporting communities to act on them. Where NGOs are not present, individuals and other concerned groups might play the same role. The inspiration for projects varies. Some are motivated by direct observation of a social or economic problem or opportunity. For example, the management of an NGO might be aware of a potential market for a local product (such as handicrafts or mangoes). By organizing the community to collect, process, and transport their products, it might be able to increase local income and employment. Other projects are based on replicating successful initiatives from elsewhere, learned through informal contact with funding agencies, other NGOs and/or explicit knowledge-sharing activities (such as workshops, publications, Web sites, and requests for proposals).

The strong network of NGOs and CBOs is associated with several positive externalities. The "social capital" accumulated within these groups, in the form of roles, norms, values, social networks, and opportunities for communication and sanction, has been found effective in making the groups more resistant to outside shocks, reliable in repaying loans, and efficient in conserving natural resources.[11,17,20]

In theory, the vetting and monitoring process should be similar to what takes place in the corporate world where investors and prospective shareholders regularly assess the financial performance of companies.

Civil Society Value Chain

The value chain that rewards NGOs for positive results is still evolving. For many smaller NGOs and those CBOs that do not engage in economic activity, the primary source of funding is private donations or grants from foundations and government agencies. Some work closely with national or regional governments. The majority rely on foundations, charities, and international NGOs that themselves raise money from governments, donors, and larger institutional sources and then grant (or subcontract) to smaller local NGOs. To establish and maintain funding relationships, NGOs must demonstrate positive social and economic outcomes and the potential for broader impact arising from their work.

Recently, CBOs have had more market-based opportunities for leveraging their social, economic, and environmental impacts. Here are a few examples:

Organic certification. Organic certification provides producer cooperatives a price premium for practicing healthier and more environmentally sustainable growing practices;

Fair trade. Fair-trade certification provides producer cooperatives a price premium to promote better compensation for farmers;

Microfinance. Microfinance groups receive preferential access to capital based on their social impact as measured by various indicators; and

Environmental services. Farmers receive payment for providing environmental services (such as retaining native trees and maintaining soil nutrients) to the global community.

To institute a fair reward system, these mechanisms require a compelling and verifiable demonstration of benefit, but the tools available to NGOs for doing so are limited. Their information systems, even when automated, are often based on manual data collection and entry, locally developed data management and reporting software, and limited IT infrastructure and expertise. In these scenarios, the quality, quantity, and timeliness of data might all be lacking. Outside the rapidly commercializing world of microfinance, NGO systems are rarely audited by donors, governments, or other agencies.

As a result, prospective funders find it difficult to assess the quality of their work. Dedicated and effective grassroots organizations have trouble differentiating themselves from superficial, media-savvy operations. Fraud and corruption abound, including sham projects receiving prestigious awards and multimillion-dollar grants.[2] In theory, the vetting and monitoring process should be similar to what takes place in the corporate world, where investors and prospective shareholders regularly assess the financial performance of companies. Even this highly resourced and regulated relationship has experienced notable failures of oversight. For social investing, ensuring due diligence is made even more difficult by limited NGO resources and capacity and by the fact that there are many different ways of defining, measuring, and demonstrating impact.

The vision and mission of any given NGO usually emanates from its leadership, often dedicated professionals with strong ties to the community. Many NGOs face significant constraints in terms of technical expertise and management capacity, particularly beyond that leadership. Still, no matter how charismatic the head man or woman might be, the field staff is the glue that binds the NGO to the community. Field staff members are recruited from villages and have a thorough understanding of the local culture and context. This is not to say their jobs are easy, anything but. Field staff must travel from village to village using unreliable public transportation or walking, even as they build and maintain relationships with local residents.

NGOs are usually organized around a specific development purpose and infused with optimism, idealism, and institutional legitimacy, making them effective change agents. On the other hand, local people may be struggling just to make ends meet. One of the field staff's many tasks is to translate the organization's mission into community terms and encourage its participation. Fostering strong communication with the community is essential for ensuring that projects make progress and address real needs.

Technologies as simple as a digital camera have been found effective for ensuring accountability.

System Requirements

The basic requirements for NGO information systems are similar to those for other organizations. Field staff members collect data using paper forms and PCs or other computing devices. This data is aggregated in a database or manual paper file. From there, it is processed by either a computer or human to generate reports useful for communicating with management, staff, beneficiaries, outside parties, and funding sources. To fulfill their objectives, NGOs must address a number of information and communication needs:

Operations. They must coordinate the operations associated with their activities. As an example, for an agricultural cooperative, quality control, procurement, processing, and delivery must all be part of their systems and processes. For those NGOs working in microfinance, loan payments must be collected, documented, and deposited in a bank. Even if a CBO is nominally in charge of these processes, unless it has sufficient scale and capacity, it often needs the assistance of a supporting NGO and its field staff;

Training and monitoring staff. NGO field staff members operate in far-flung areas with limited supervision from management. They often assist and even manage CBO operations. They are the source of feedback from the community, including problems, needs, and results of ongoing efforts. Field staff must be trained to execute these responsibilities and monitored to ensure they do not cheat, shirk, or introduce bias. Given their low pay and limited training, the difficulty of traveling in rural areas, and opportunities to leverage their NGO affiliation, this is not easily done. However, technologies as simple as a digital camera have been found effective for ensuring accountability[5];

Documenting results. Most staff data-collection activity is for documenting the results of NGO activities. The data could be used for internal monitoring and learning but is more often motivated by the reporting demands of funders. Traditionally, the funders request quantitative indicators (such as increased income, loan repayment rate, or number of people treated or vaccinated). Because these indicators favor preordained and externally determined metrics of success, more community-driven ways of measuring results have also been developed. One such method asks beneficiaries to report the "most significant change" associated with an

Coffee producers of Aldea Cocolá Grande, Barillas, Guatemala. (Photo by Baltazar Francisco Miguel, general manager of ASOBAGRI, an organic coffee cooperative in Barillas, Guatemala.)

intervention through qualitative interviews and participatory video tapings (www.mande.co.uk/MSC.htm);

Accounting. NGOs must show how they've allocated expenses to specific projects for reporting to funders. As many smaller NGOs lack funding for their infrastructure and core operations, managing this overhead requires finding ways to account for these expenses through funded projects. Most mid-to-large-size NGOs use some kind of commercial accounting software to manage their books. Most registered entities also seek the services of certified accountants. Specific country regulations for nonprofits (such as for filing taxes and/or documenting foreign donations) must be followed;

Reporting. Impact and budget data must be reported regularly to donors, governments, and/or regulatory agencies. Each usually insists on using its own reporting formats and indicators. NGO information systems, from data collection to reporting, are driven by these requirements and built and modified as they arise, with little upfront planning or dedicated design effort. The result is a hodge-podge of code and data that is impossible to maintain—and ultimately might not be secure or usable;

Decision making and learning. The focus on external reporting leaves little room for internal decision making and learning. Many NGOs do not employ a systematic effort to collect data useful for understanding the real effect of their programs. As a result, many NGO managers run their organizations through personal intuition and by following trends, informed by whatever knowledge reaches them through their organization's ad hoc communication channels. Only large professional NGOs or those associated with international networks that are able to leverage significant economies of scale have custom-built information systems designed to meet their own requirements; even these usually meet only the needs of NGO headquarters, not those of their regional or field offices;

External knowledge. NGOs serve as a gateway for accessing external information useful to CBOs and rural communities. For agricultural cooperatives, this might include weather forecasts, reviews of new seed varieties, and descriptions of new farming practices and technologies. NGO management and staff must filter, adapt, and/or translate it so it is relevant for and understandable by the community. Ideally, the community should be able to contribute its knowledge as well; it, too, must be filtered, adapted, and/or translated by the NGO to tailor it to a broader audience; and

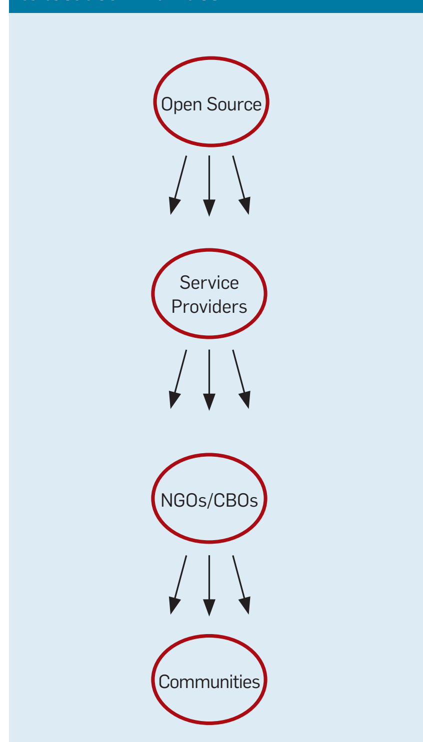
Community participation. The most basic operational task for NGOs and CBOs is maintaining consistent bidirectional communication with community members and is especially important for the governance of membership-based organizations like cooperatives. While field staff members are rewarded for spreading the NGO's message and collecting data required by funders, there is little incentive for them to report unsolicited feedback from the field. As a result, many NGO managers have only a limited understanding of local problems and the status of their own organization's field efforts. Rich information dissemination and feedback tools could help NGOs be more accountable to their communities and encourage greater community participation.

Promising Approaches

Rural NGO and CBO information and communications needs (see Figure 1) are supported by a number of promising technology developments:

Mobile phones. The decreasing cost and improving capabilities of mobile

Figure 1: Information services ecology for rural communities. Open source software enables technology service providers to support NGOs and CBOs that provide relevant and accessible information services to local communities.



phones make them a compelling computing platform, especially for mobile field staff. Beyond voice and text communications, they might also be used for data collection, decision support, and knowledge dissemination, including the capture and playback of multimedia. My students and I have implemented mobile applications in India, Mexico, and Tanzania for documenting microfinance transactions,[13] inspecting organic coffee farms,[18] and diagnosing and treating childhood illnesses.[3] One constraint we continue to face is the lack of open, accessible, cross-platform mobile development tools;

Open source. Implementing business software requires extensive customization, training, and maintenance. Providing these services is the basis for successful service business models for such companies as IBM, Oracle, and SAP. However, it would be incredibly difficult for a centralized organization to provide these services to remote locations across a large geographic area.[12] Local service providers, operating in capitals and regional hubs, are essential to the NGO information services ecology. Open source software could provide a foundation upon which other providers could build customized applications and/or provide value-added services;

Cloud computing. Due to the challenge of installing and maintaining software in remote locations, a software-as-service approach might also be useful.[7] Software in the cloud would be easier to install, maintain, and update. However, in the rural developing world, the lack of access to reliable broadband connectivity limits uptake of this model. Improving the quality and availability of high-bandwidth network connections could alleviate this limitation; so would providing a mechanism for offline access. Even so, localization and other customizations require an open and flexible software architecture;

Store-and-forward networking. Due to spotty network coverage, asynchronous networking technologies to provide offline access to services are useful for PC-based and especially for mobile clients. This requires methods for intelligent caching and transfer of data. Given that NGO staff members regularly visit both field and head offices, these offices can serve as asynchronous communications hubs for rural communities. Highly decentralized projects in remote regions with low population density and limited infrastructure, and where field staff may stay in the field weeks at a time, could benefit from multihop networking approaches[4];

Many NGO managers run their organizations through personal intuition and by following trends, informed by whatever knowledge reaches them through their organization's ad hoc communication channels.

Long-distance wireless. For applications requiring synchronous communications (such as telemedicine and voice telephony), the decreasing cost of long-distance, high-bandwidth wireless connections is a promising development.[15] They represent a compelling tool for linking remote NGO offices with human and information resources that are likely available only in cities and other centralized locations;

Narrowcast media. The declining cost of media production, dissemination, and playback over the next decade promises to revolutionize how NGOs disseminate information. Community radio is a powerful yet low-cost tool for broadcasting audio content in a specific geographic area. Using video cameras and DVD players to generate and distribute media content has been found beneficial for activities as diverse as children's education and agricultural extension.[6, 21] How to enable communities to generate their own content or provide feedback to content producers is an important and largely ignored challenge.

Bootstrapping a Service Ecology

Implementing computing applications for rural NGOs requires building local capacity for software development, integration, customization, installation, training, maintenance, and support. In most developing countries, the IT services sector, if there is one, focuses on outsourcing and building commercial applications. NGOs usually rely on second- and third-tier programming talent. For large development projects, the only alternative is to hire expensive foreign consultants or companies from established IT markets and fly them in and out as needed. For many NGOs this is not feasible financially or logistically. As a result, most applications intended for NGOs are being developed by poorly trained local programmers using a limited set of tools.

Several approaches are required to address this problem. Foremost is the emergence of a robust IT services industry in more countries. Working in Africa and India, I have personally met many talented developers and small software companies committed to working with NGOs. More opportunities and resources must be provided by governments and funding agencies to encourage them to serve the nonprofit sector. This could help bootstrap IT service businesses in countries with limited access to outside markets. Enabling these relationships requires a better social network connecting NGOs, software-training institutes, fledgling software companies, and funding sources, including (social) venture capital.

Better software-development tools would also help, particularly for mobile applications. Microsoft's .NET framework is a well-integrated, well-documented, full-featured proprietary development platform for which there is significant infrastructure worldwide for developer training and support. However, many NGOs are unwilling to pay license fees for software and uncomfortable using pirated software and may oppose being tied to a particular operating system. As a result, there is a notable push by NGOs toward open source. For mobile applications, more openness and standardization are sorely needed, as is better support for cross-platform development. Recent efforts by Apple, Google, and Nokia to provide more open development platforms seem to be a step in the right direction.

In terms of application software, neither the shrink-wrap nor the proprietary-software-plus-services model has worked for rural NGOs.[12] Providing a suite of useful open source applications could be an effective alternative, potentially having a cascading positive effect on local economic development. Each open source application could create a number of opportunities for local IT companies to serve NGOs, each in turn serving a number of local communities and individuals. This is the kind of public good that governments and donors should provide.

Several open source applications have been developed for NGOs, including for managing microfinance programs (www.mifos.org) and maintaining electronic health records (www.openmrs.org). This trend allows local software providers to focus on adding value, including customization, localization, installation, maintenance, and support. Improvements and additions to the code can later be fed back to the main branch, lowering the barrier to entry for future implementers. Working on such globally distributed projects requires effective collaboration-and-governance tools that work across time zones and cultures.

Finally, the global research and development community must address the challenges of computing in the rural developing world, one of the last frontiers of computing adoption. Researchers working in this field have already provided novel insights into shared computing, interfaces for semiliterate users, and long-distance networking technologies for reaching remote locations.[14, 15, 16]

Working Systems

To illustrate these concepts, I describe two systems—Self-Help MIS and Digital ICS—my students and I built in collaboration with rural NGOs and CBOs. These systems demonstrate how information technology is being used to automate existing processes and provide new opportunities for development. While a formal long-term evaluation is pending, initial results have been positive and generated additional demand from other civil society organizations. (For more, please see [13, 18]).

Self-Help MIS. Self-help groups, or SHGs, are semi-independent microfinance groups located throughout rural and peri-urban India. They usually consist of 15 to 20 members, mostly women (see Figure 2). Group members save money during regular group meetings, rotating the accumulated capital in the form of loans that are repaid with interest, increasing the group's corpus. When the total amount becomes too much for the SHG to manage, it is disbursed back to the group members. The cycle then starts again, sometimes with the reshuffling of members among groups. SHGs are promoted by NGOs, banks, farmer groups, individuals, and government agencies in India. Beside the immediate benefit of providing access to financial services, SHGs are also a mechanism for organizing other community development activities.

Working with SHGs is attractive for local banks, as they lessen the need to build physical branch offices in remote areas and thereby reduce the cost of providing financial services. However, banks often have no way to address SHG quality and history, making it difficult to determine the risks associated with various products and services (such as the lending of money). Due to limited literacy and education, many SHGs keep inconsistent and/or incomplete records. Many need the help of literate people, sometimes the field staff of a local NGO, to maintain their records

Figure 2: Field staff capturing data from a self-help-group meeting.

and prepare reports for internal oversight and external assessment. Lack of transparency and internal controls can also lead to problems within an SHG, including the capture of assets by elite members by stealing and/or taking a disproportionate share of loans.

In collaboration with ekgaon technologies (or "one village" in Hindi, www.ekgaon.com), a technology firm I cofounded in 2002 in India, and the Covenant Centre for Development (www.ccd.org.in), an SHG-promoting NGO in Tamil Nadu, India, I developed Self-Help MIS, a system for monitoring a network of SHGs. Group members record transaction data using paper forms that are processed by field staff using a mobile phone to capture and upload data to a database application running at the NGO office. When field staff visit again, they print reports to take back to the SHG, which uses them in its next meeting and as supporting documentation when applying to a bank for a loan. NGO management can also use this data to monitor SHG performance, addressing potential trouble spots and demonstrating the outreach and impact of the SHG program to their funders. If new services or products are introduced, NGO management can empirically determine their uptake and impact.

Deployed since October 2006, Self-Help MIS is being used by more than 1,000 SHGs and their 15,000 members. Ekgaon has negotiated a service contract to translate the system into six languages (besides the original Tamil) and deploy the system in several additional states. As one of the first SHG information systems to provide intragroup transparency through direct field-level data collection, one World Bank project is using it as a model for learning how to establish a network of SHG credit reporting bureaus, with a potential total outreach of millions of group members.

Digital ICS. My students and I are working with two farmer cooperatives in Mexico and Guatemala that produce certified organic and shade-grown coffee (without chemicals while retaining native trees on coffee parcels). Maintaining these certifications requires arduous paperwork and diligent internal oversight. In both cooperatives, they are inspected every one to three years by various certifying agencies. External inspectors visit a randomly selected cross-section of farm plots and processing equipment. If they observe inconsistencies, the cooperative's certification (and price premium) might be at stake.

To avoid this risk, cooperatives institute stringent internal controls. Twice a year, local inspectors, selected from the ranks of field staff, office staff, and trusted farmers, visit every coffee parcel belonging to each cooperative's members. This inspection requires completing and delivering a lengthy form to the cooperative's internal control manager. The manager reviews the inspection data, singling out farmers having problems with the transition to organic chemical-free production or at risk of violating certification requirements. On their next visit, field staff convey advice, reprimands, and/or sanctions to farmers on behalf of the manager.

To automate this process, my students and I implemented the Digital Internal Control System (or DigitalICS, pronounced *digitalix*). Inspectors fill out forms directly on Nokia 6600 mobile phones, collecting pictures of farm plots and equipment and audio recordings of farmers' explanations, questions, and comments that are useful as evidence of breaches of certification requirements, for resolving disputes, and for targeting agricultural training and extension activities. As evidence of their visits, the cooperative requires that each inspector photograph the farmers on their plots (see Figure 3). This is intended to ensure that staff visit and document every parcel, not easily done when some take hours to reach and are on steep inclines miles from the nearest road. On the next visit to the cooperative office, inspection data is uploaded to a Web-based database application we developed. The internal control manager reviews inspection forms through this application, automatically generating reports for certifying agencies. The manager also enters feedback and notes for later discussion with field staff and for conveying back to the farmers.

The same technology and process can be used to provide traceability, grading, collecting feedback from farmers, monitoring social and economic impact, and instituting reward mechanisms tied directly to farm-level quality. Digital ICS is one of the first systems to provide these features in an integrated manner by allowing for the collection of rich multimedia data through mobile phones. Digital ICS has been through two trials conducted with the Coffee Grows Association of Oaxaca, Mexico (CEPCO), where it is being used by more than 1,000 farmers as part of their regular operations. In the future, CEPCO and Asobagri (our partner in Guatemala) would like to use the system to also produce better marketing materials and facilitate more communication between producers and consumers. Another idea is to monitor the environmental services provided by coffee farmers, including retaining native trees and/or planting new ones. After being verified by a third-party process similar to certification, these services can be sold as credits on the international carbon market.

Figure 3: Photo used as evidence that field staff visited a particular farm parcel.

Development as Engineering

Grace Hopper, a computing pioneer in the U.S. Navy, wrote that the role of computers is "freeing mathematicians to do mathematics."[9] The focus of today's efforts to develop information systems that support rural development is similar: free development practitioners to do development. Development economists have largely given up on theory-driven models, increasingly recognizing that there is no single magic elixir for development; solutions to specific problems must be conceived, implemented, rigorously evaluated, replicated, and scaled.[1] The development process has become more incremental and empirical, like engineering, the process of identifying, validating, and generalizing solutions to problems.

Herbert Simon, a professor at Carnegie Mellon University, described design and engineering as the "sciences of the artificial."[19] Unlike the natural sciences, which are concerned with how things are, these fields are concerned with how things should be. The goal of an engineer is to build systems that achieve some desired result, based on a process that centers on the generate-test cycle, producing likely solutions and testing whether they are successful relative to some desired metric. How to design in anticipation of a positive result and evaluate whether or not that result is as anticipated are the methodological and empirical bases for this trade.

From this perspective, ICTs should not be viewed as a panacea for development but as a tool allowing local change agents (like NGOs) to be more effective and accountable. As the examples here show, computing is able to support local organizations in generating and sharing new ideas, implementing and disseminating projects, and measuring their effects. The goal is to facilitate a virtuous cycle of experimentation by augmenting local efforts and improving their visibility and accountability to the institutions that might support, replicate, scale and/or learn from them.

Here, I have highlighted the role of NGOs, but one could equally focus on government agencies and local businesses. NGOs are important sources of innovation for rural development. Their familiarity with the problems and people of a region, along with their idealism and activism, enable them to identify and pursue novel development strategies. Their lack of government and commercial accountability allows them to be more creative and free in this enterprise. If an idea is demonstrably successful, it can later be scaled through state or business approaches.

Conclusion

How can computer scientists and engineers design computing technologies to achieve desired development objectives? Mobile phones have demonstrated a positive contribution, improving the efficiency of commodity markets in a variety of contexts.[10] While the simple ability to communicate provides an immediate advantage, the effect of computing often takes time to realize. Specifically, it must be accompanied by organizational and procedural changes that take advantage of the new opportunities computing affords (such as decision making based on analysis of past performance).

Many technologies useful for achieving these goals are technologically simple. Complexity arises in their adoption and interaction with human social and organizational systems. To manage this complexity, computer scientists and engineers must improve the process of designing, introducing, and evaluating computing technologies. This engineering process involves understanding local users, designing appropriate and easy-to-introduce solutions, and demonstrating their ability to achieve beneficial social and economic change. This process must also be contextual and iterative, requiring rich learning from field tests and pilot deployment of varying scale and duration.

Approaching these issues rigorously should require methods from the social sciences, including user studies and ethnography used for decades to study human-computer interaction. The emphasis on process must now go beyond user interface design to encompass end-to-end application and systems design. Students of computing must not only learn the mechanics of computing and its underlying abstractions and analytic foundations but also the how to design and evaluate computing systems that achieve specific human objectives.

Acknowledgment

Many thanks to my NGO and CBO field partners, including Asobagri, CEPCO, CCD, Jatan Trust, and the Mahakalasm SHG federations, for allowing me to work with and learn from them. Thanks, too, to my wonderful students, including Yaw Anokwa, Kuang Chen, Brian DeRenzi, Neil Patel, and Yael Schwartzman, for sharing their insight and experience from the field, and to Neal Lesh, Deborah Estrin, and the anonymous reviewers for their valuable comments on earlier drafts.

References

1. Banerjee, A. Inside the machine: Towards a new development economics. *Boston Review* (Mar. 2007); bostonreview.net/BR32.2/banerjee.php.
2. Behind the scenes of Shidhulai Swanirvar Sangstha. *Star Weekend Magazine* (Sept. 2007).
3. DeRenzi, B., Lesh, N., Parikh, T., Sims, C., Maokla, W., Chemba, M., Hamisi, Y., Hellenberg, D.S., Mitchell, M., and Borriello, G. e-IMCI: Improving pediatric health care in low-income countries. In *Proceedings of the 26th annual Conference on Human Factors in Computing Systems* (Florence, Italy, Apr. 5–10). ACM Press, New York, 2008, 753–762.
4. DeRenzi, B., Anokwa, Y., Parikh, T., and Borriello, G. Reliable data collection in highly disconnected environments using mobile phones. In *Proceedings of the SIGCOMM Workshop on Networked Systems for Developing Regions* (Kyoto, Japan, Aug. 27). ACM Press, New York, 2007.
5. Duflo, E. and Hanna, R. Monitoring works: Getting teachers to come to school. *NBER Working Papers* (Dec. 2005); titsearch.ssrn.com/sol3/papers.cfm?abstract_id=895584.
6. Gandhi, R., Veeraraghavan, R., Toyama, K., and Ramprasad, V. Digital green: Participatory video for agricultural extension. In *Proceedings of the Conference on Information and Communication Technologies for Development* (Bangalore, India, Dec. 15–16, 2007), 21–30.
7. Gruman, G. and Knorr, E. What cloud computing really means. *InfoWorld* (Apr. 2008).
8. Hardin, G. The tragedy of the commons. *Science 162*, 3859 (Dec. 1968), 1243–1248.
9. Hopper, G.M. The education of a computer. In *Proceedings of the 1952 ACM National Meeting* (Pittsburgh, PA). ACM Press, New York, 1952, 243–249.
10. Jensen, R. The digital provide: Information (technology), market performance, and welfare in the South Indian fisheries sector. *Quarterly Journal of Economics 122*, 3 (Aug. 2007), 879–924.
11. Karlan, D.S. *Social Capital and Group Banking*. BREAD Working Paper No. 62 (Apr. 2004); www.cid.harvard.edu/bread/abstracts/062.htm.
12. Parikh, T.S. Rural microfinance service delivery: Gaps, inefficiencies and emerging solutions. In *Proceedings of the Conference on Information and Communication Technologies for Development* (Berkeley, CA, May 25–26, 2006).
13. Parikh, T.S., Javid, P., Kumar, S., Ghosh, K., and Toyama, K. Mobile phones and paper documents: Evaluating a new approach for capturing microfinance data in rural India. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal, Canada, Apr. 24–27). ACM Press, New York, 2006, 551–560.
14. Parikh, T.S., Ghosh, K., and Chavan, A. Design studies for a financial management system for microcredit groups in rural India. In *Proceedings of the 2003 Conference on Universal Usability* (Vancouver, Canada, Nov. 16–17). ACM Press, New York, 2003, 15–22.
15. Patra, R., Nedevschi, S., Surana, S., Sheth, A., Subramanian, L., and Brewer, E. Wildnet: Design and implementation of high-performance wi-fi-based long-distance networks. In *Proceedings of the Fourth USENIX Symposium on Networked Systems Design and Implementation* (Cambridge, MA, Apr. 11–13, 2007), 87–100.
16. Pawar, U.S., Pal, J., Gupta, R., and Toyama, K. Multiple mice for retention tasks in disadvantaged schools. In *Proceedings of the Conference on Human Factors in Computing Systems* (San Jose, CA, Apr. 28–May 3). ACM Press, New York, 2007, 1581–1590.
17. Pretty, J. Social capital and the collective management of resources. *Science 302*, 5652 (Dec. 2003), 1912–1914.
18. Schwartzman, Y. and Parikh, T.S. Using CAM-equipped mobile phones for procurement and quality control at a rural coffee cooperative. Presented at the WWW 2007 Workshop on Mobile Web in the Developing World (Banff, Alberta, Canada, May 8, 2008).
19. Simon, H.A. *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 1969.
20. Upho, N. and Wijayaratna, C.M. Demonstrated benefits from social capital: The productivity of farmer organizations in Gal Oya, Sri Lanka. *World Development 28*, 11 (Nov. 2000), 1875–1890.
21. Wang, R., Sahni, U., Sobti, S., Garg, N., Singh, J.P., Kam, M., Krishnamurthy, A., and Anderson, T. *The Digital Study Hall*. Technical Report TR-723-05. Department of Computer Science, Princeton University, Princeton, NJ, Mar. 2005. dsh.cs.washington.edu/StudyHall/.
22. World Bank and NGOs; library.duke.edu/research/subject/guides/ngo_guide/igo_ngo_coop/ngo_wb.html.

Tapan S. Parikh (parikh@berkeley.edu) is an assistant professor in the School of Information at the University of California, Berkeley, and has an affiliate appointment in the Department of Computer Science and Engineering at the University of Washington, Seattle.

DOI:10.1145/1435417.1435434

They promise mobile information access and assistive Web services for tens of millions worldwide.

BY DAWN N. JUTLA AND DIMITRI KANEVSKY

*wise*Pad Services for Vision-, Hearing-, and Speech-Impaired Users

OUR VISION FOR the *wise*Pad system, named for the iconic three wise monkeys that see no evil, hear no evil, speak no evil, is a full-service computing platform designed to deliver personalized image-, audio-, and text-transformation services to people with impaired vision, speech captioning for people with hearing problems, and language-translation, text-summarization, and pictographic-illustration services for people needing help with both spoken and text-based language.

*wise*Pad services employ a generic head-mounted display content-transformation device of the type described by Kanevsky and Sorenson.[7] The device captures image(s) on an output device (such as iPod screen, PC monitor, and projection screen) and receives multimedia signals input from other channels. Related *wise*Pad services then use the service's content-transformer functions to translate captured or received images and display them on the screen of the device.

*wise*Pad transforms an image in several ways: For the vision-impaired user, it magnifies parts of the original image to facilitate visual perception and thus comprehension. For the hearing-impaired, it displays closed-captioned video, pictures, and text in the lenses of the user's eyeglasses. And for speech aid, it might transform text into simplified text in the same language, speech into another language, speech into summary form, and/or speech by adding pictographic icons intended to help in language understanding in the same way pictures and icons help young children learn to read.

We expect within the next 10 years, *wise*Pad will deliver an unprecedented level of personalization for its users, including for artifact selection, page ordering, language, color, magnification, and image-feature transformation. *wise*Pad will be able to store these user choices as preferences in a customized software "skin" in its user interface.

*wise*Pad will be available in several implementation versions, including ordinary-looking glasses with a frame large enough to hold a tiny USB port, miniature transformer, display chips, and goggle-style eyewear with wireless microphones supporting audio output and input. Users select the version that best suits their needs.

Figure 1 shows the kind of eyewear adapted for *wise*Pad services as an output device. Lumus Ltd., Rehovot, Israel (www.lumusvision.com), designs, manufactures, and markets a lightweight designer frame with twin microdisplays at full VGA-type resolution. Tiny projectors embedded in the eyewear's stems project images with a 27.5-degree field of view, as if the wearer were "watching a 70-inch

Future transparent wearable display (www.microvision.com).

PHOTO COURTESY OF MICROVISION INCORPORATED

TV two-to-three meters away" (www.lumusvision.com); users see the images reflected on the lenses. Alternatively, an IBM belt-wearable portable computer (www.research.ibm.com/WearableComputing/) with built-in GB-capacity hard drive, lightweight eyewear, and tactile controller[7] delivers images captured through an output device, transforming them and projecting revised images onto the lenses. Microvision, Inc., Redmond, WA (www.microvision.com/wearable.html), designs, manufactures, and markets generic services for its transparent wearable displays, including to give users the ability to view their email and caller ID information without first retrieving their mobile devices while also being able to "see" GPS directions in the lenses of their glasses.

For the Visually Impaired

A 2003 Forrester Research report[8] concluded that 60% of working-age adults in the U.S. (approximately 101.4 million people) would likely benefit from accessible technology following a disability or impairment. In 2003, 27% of working-age adults in the U.S. (approximately 45.9 million people) had some sort of visual impairment. Meanwhile, by 2020, 20% of working-age adults in the U.S. (approximately 64 million people) will be 55 or older[9]; aging and its attendant susceptibility to sight-impairing diseases (such as diabetes) will increase demand for accessible technology to help them stay independent and productive.

They'll likely need text and image magnification, color-modification, and usable navigation and information-location services. An example of how *wise*Pad might deliver such a service is when vision-impaired Web users seek out the search window on a Web page when browsing a particular site. Even locating a search window can take such a person much longer than it takes a fully sighted person. Thus *wise*Pad promises the ability to display a search window in the user's focal area, magnify the retrieved information, and/or transform text by summarizing or translating the search results before displaying it in the user's predetermined area of focus. The idea is for users to look through their glasses to the page and for the search window to pop up automatically. For an automated service to support quick recognition of an artifact (such as a search window) on a Web page, the service's search-window component would have to be able to identify itself programmatically through a semantic language tag.

Transforming text into speech would also greatly benefit people with a visual impairment. For example, IBM's ViaScribe[10] is bundled with such a service (www.liberatedlearning.com/technology/index.html). *wise*Pad will, as its constituent technologies and markets mature, similarly offer such a service for visually impaired readers of business reports, consumer books, and other media.

Ubiquitous Participation

The most recent (2006) demographic data available from the U.S. National Center for Health Statistics (www.cdc.gov/nchs/) concluded that 17% of the adult population in the U.S. (approximately 36.5 million people) have some kind of hearing problem. Gallaudet University Research Institute's 2006–2007 survey (gri.gallaudet.edu/Demographics/2006 National_Summary.pdf) sampled 37,352 U.S. students (most 6 to 17 years of age) with some sort of hearing impairment. In the range of hearing loss, from mild to profound (from 27dB to 91dB and above), the profound-loss category characterizes more than one-third of these students. Moreover, of the surveyed students 8.1% reported having no access to support services, 57.5% had access to speech training and therapy services, 8.8% had access to itinerant teacher services, and only 0.9% had access to real-time captioning services.

Captioning is one of the most useful technology-based services for people with diagnosed hearing disabilities. Experiments with university students (2000–2008) in Australia, Canada, Japan, New Zealand, and the U.S. using ViaScribe showed positive learning results in terms of enhanced test-taking performance for the disabled (as well as for the nondisabled, as they also appreciate the value of captioned notes) due to increased student participation in class (better note taking) and understanding of spoken and assigned material.

Many hearing-impaired people would like to be able to watch in-flight movies, as well as home videos, and generally participate more fully in classroom, workplace, and social environments. *wise*Pad closed-captioning services on wearable-display eyewear would help them by reading the words they are unable to hear. In 2002, in order to implement the *wise*Pad closed-caption service, Basson and Kanevsky[1] proposed a system to process a signal containing closed captions for audio content associated with multimedia (such as podcasts, videos, and text). In a variety of situations (such as an e-commerce transaction) where some amount of time delay is acceptable, a human-based transcription service could generate captions in real time. When time delay is less tolerated (such as in a meeting), caption accuracy might be sacrificed for real-time automated transcription (such as through ViaScribe and Nuance's Dragon Speaking, www.nuance.com/naturallyspeaking/). These tools deliver accurate text, depending on language domain (such as health and education), acoustic environment, and voice-model quality.

Figure 1: *wise*Pad services use designer eyewear.

Multiple Languages

The Gallaudet Research Institute's 2006 survey (gri.gallaudet.edu/Demographics/2006_National_Summary.pdf) also reported that over 50% of the 37,352 U.S. students (ages 5 to 18) with hearing disabilities (or almost 19,000 children) suffer from additional impairments possibly involving vision, learning, and attention-deficit disorder. *wise*Pad aims to provide services for users with multiple disabilities, as well as for students (both children and adults) whose first language is not English, young readers, and students with learning disabilities and low literacy levels.

Helen Keller (1880–1968), a famous inspirational scholar born in Alabama in the U.S., was deaf and blind as a result of a brief illness with a high fever (likely scarlet fever or meningitis) when she was two years old. As a young child she learned first that she could communicate through touch and gesture. It was later that she understood that objects had names and that words could be used to describe them. Position detectors, movement tracers,[5] and other technologies allow consumers to program gestures into their everyday devices (such as wristwatches, automobiles, and MP3 players) to quickly produce output (such as display a timer, scroll down a window, and select a particular song) without having to touch a button. A video posted on YouTube called "Captioned Version of Artificial Passenger" (www.youtube.com/watch?v=APLEwmPBeoA&feature=channel) demonstrates the technologies described by Kanevsky and Zlatsin[5] now incorporated into *wise*Pad by the author Kanevsky and his colleagues at IBM T.J. Watson Research Laboratory. These gesture-based technologies allow users to "train" the behavior of electronic devices (such as a mobile phone camera) to respond to user-taught gestures. *wise*Pad combines voice sounds and text through these gestures. The various *wise*Pad user interfaces for obtaining services can be as innovative as a user's imagination can make them. For example, a user can program a gesture (such as touching a hand to a face) in a *wise*Pad-enabled device to add pictures to words, as well as dictionary meanings and synonyms.

With them, a user would be able to transform a page of text and display a summary translation with pictographs on the lens. Users would also be able to train *wise*Pad-enabled glasses to support their own idiosyncratic gestures; for example, tapping the glasses frame twice might connect the user to the Web to view email on the lens display; a user-trained rotation of the frame might magnify visual content; and composite geometric gestures (touch and/or touchless) might adjust the colors in the display.

The *wise*Pad system could also employ Infoscope-type services[3] to, say, copy an image of a menu written in Chinese, communicate it through the Internet for translation, and then display a translated menu in English on a user-display device (such as the user's glasses). Language translation in *wise*Pad also acts on audio input. Airplane-based or other mobile transport servers might thus be able to host automatic translation software[4,6] for train, air, and automobile travelers using *wise*Pad. Alternatively, translation services might be delivered through the Internet for those applications that are able to tolerate several seconds of delay.

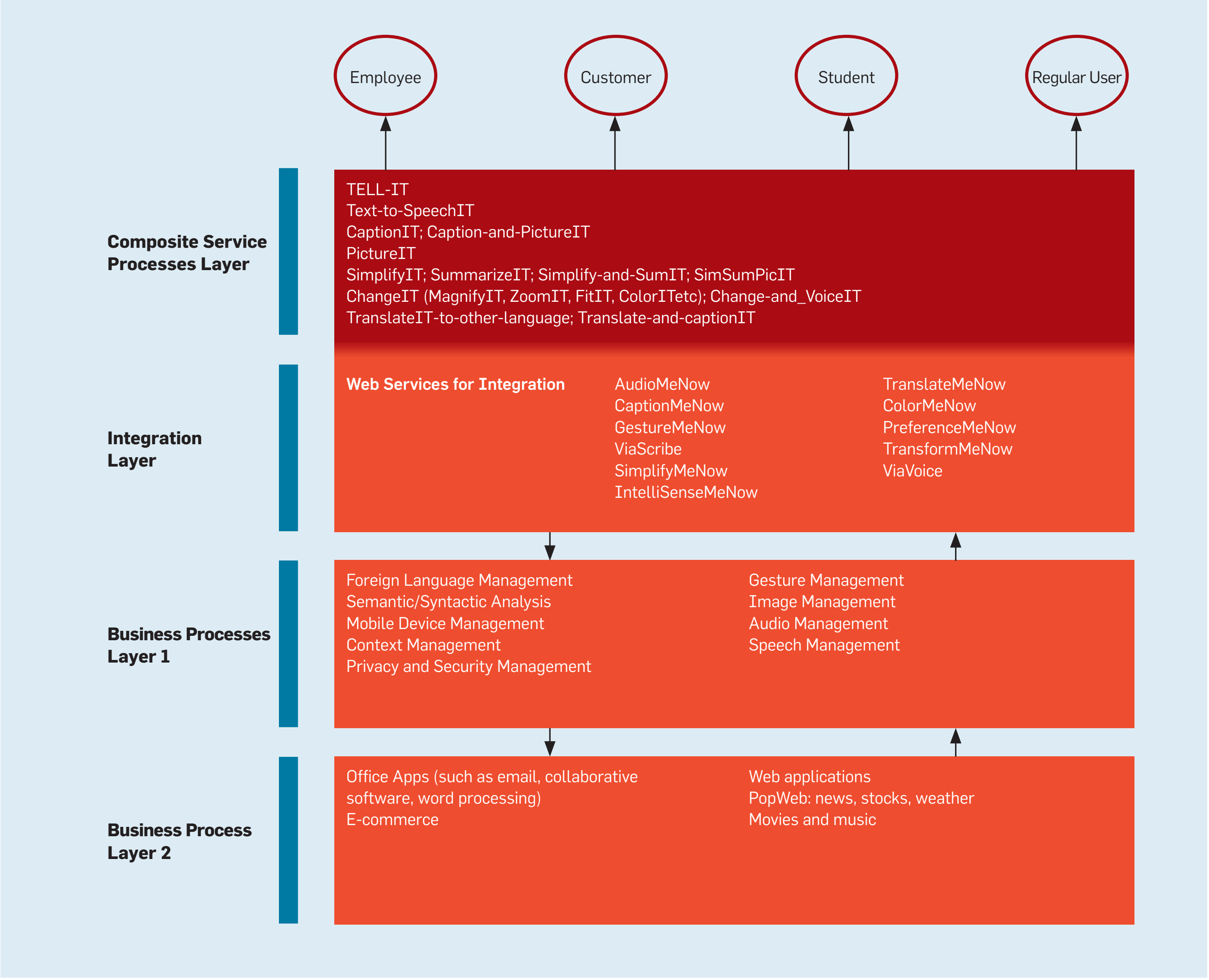
Others who might find *wise*Pad services useful are newly arrived immigrants or visitors to North America who read and write English better than they speak and understand it. For people with some kind of hearing impairment, closed captioning is a proven way to enhance communications. Children and other users who need literacy aids would also benefit from *wise*Pad's integration of pictographic illustration within the captioning it generates in the user's display. Moreover, document and partial-text summarization would be available through *wise*Pad implemented (pending negotiation with other patent holders) with the help of methods like those in Fein et al.,[2] including word- and phrase-frequency-based statistical analysis and cue-phrase analysis.

Blueprint

In 2007, we designed the *wise*Pad service-oriented architecture (SOA)-based blueprint (see Figure 2) to express business logic in terms of services that use XML to describe input and output, facilitating integration of IT-application components, including transcribing speech and translating text. Integration is achieved by representing the business logic of the various application components as Web services; the table here lists the composite Web services *wise*Pad provides to its users.

These services consist of multiple layers. For example, at the portal level, users receive Web-enabled services (such as CaptionIT, generating captions from aural input) from a composite Web service process layer. Other services (such as CaptionandPictureIT, generating captioned text with symbols and pictures) consist of "horizontally" integrated component services available from the integration layer below. CaptionandPictureIT combines select-

Figure 2: *wise*PAD services blueprint.



ed dedicated logic from the underlying CaptionMeNow, TransformMeNow, and PictureMeNow service suites.

Today, the integration layer in Figure 2 consists of IBM Web services that recognize and incorporate IBM's dedicated business logic design for captioning multimedia, translating languages, and simplifying text. IBM code-named the integration layer service groupings "X"MeNow in several IBM first-of-a-kind projects (such as CaptionMeNow) that are funded in part by user organizations. However, another future option may be that existing services by other vendors (such as Verizon's and AOL's AIM and IP-Relay Services, www.ip-relay.com/sp/) could be integrated into this layer. The value of high-level integrated services would be a convenience to users with disabilities, as well as to their communicating partners.

The Business Processes Layer 1 in the *wise*Pad SOA includes specialized software modules for image processing and speech processing and other such functions. They provide the technical engine (novel scientific methods and algorithms developed and patented by IBM) for the Web services identified in Figure 2. The Business Process Layer 2 includes typical *wise*Pad applications (located anywhere on the Web) that may be used as input to the *wise*Pad service suite.

Business Model

Organizations in the entertainment, travel, retail, finance, and government sectors are all likely to be among the first paying subscribers, enabling their employees and customers to use *wise*Pad services. Being first movers will follow their strong inclusive mission statements and/or U.S. federal legislation (such as the Americans with Disabilities Act and Section 508 of the Rehabilitation Act of 1973) mandating inclusion of all demographics, physical handicaps, and employee and customer categories. Profit and productivity would be important considerations for subscribing organizations expecting to benefit through increased Web use by users and customers, especially in terms of e-commerce and related revenue.

Today, the most basic infrastructure components for the *wise*Pad architecture and services is in place, including Web services standards, popular end-user mobile Internet devices, Internet content delivery, telecom carriers, Web companies (such as Akamai.com, and Amazon.com), and the *wise*Pad architecture and its bundling of end-user services. Companies designing network architectures, including those related to *wise*Pad (such as Aircell, www.aircell.com/), deliver high-speed wireless broadband to aircraft, allowing users to send and receive email and engage in e-commerce while in flight. It would thus be possible to provide *wise*Pad services on planes, trains, ferries, and other moving vehicles. All travelers would be able to enjoy multimedia transformed by *wise*Pad to include captions in multiple languages (such as Arabic, English, French, Italian, German, Japanese, and Spanish) and/or pictographs, summaries, color-coded figures, and modified Web-page components. Mobile vision-, hearing-, and/or speech-impaired financial customers would be able to more fully participate in financial markets, shop for insurance, and do their banking on the go. Mobile retail and government online channels would also be included, as more and more *wise*Pad users browse and buy online.

Future *wise*Pad service providers, including IBM, pending negotiated agreement, must still come up with a viable *wise*Pad-related revenue model. Although revenue models differ by industry and organization, a model based on a flat subscriber fee per user per month might be an option for a technology-service vendor. Initially, on-demand models are not likely to be a good choice for providers of early-stage innovative services like *wise*Pad, as the number of subscriptions does not increase as quickly as in conventional business services (such as sales force automation) and collaboration services (such as Microsoft Office Live). However, when volume subscriptions of *wise*Pad-type services do occur—a reasonable expectation over the next 10 years in light of today's worldwide impaired-user demographics—less expensive on-demand revenue models will be possible for *wise*Pad services.

Tapping the glasses frame twice might connect the user to the Web to view email on the lens display; a user-trained rotation of the frame might magnify visual content; and composite geometric gestures (touching and/or touchless) might adjust the colors in the display.

Conclusion

The vision-, hearing-, and/or speech-impaired user is the foremost stakeholder in this *wise*Pad vision. Many of even the

Figure 3: The author Kanevsky, a deaf research scientist, testing IBM's prototype head-mounted display device with IBM's ViaVoice transcription services combined with a human-in-the-loop live transcription service from CaptionFirst, Inc. (www.captionfirst.com/), the first time transcription services were coupled with eyewear and a belt-worn portable computing device.

most technology-savvy ones are still waiting for critical enabling technologies (such as *wise*Pad) to inexpensively and transparently assist their enjoyment and ubiquitous participation in family, personal, and work activities.

Since IBM inventors, engineers, and developers unveiled their first prototype of an eyewear-attached portable computer (www.isrl.uiuc.edu/~chip/projects/timeline/1999jones.html), many talented people both in and outside of IBM have participated in creating the technologies needed to make the *wise*Pad services system a reality. They have since championed and advanced the standards and technologies for Web 2.0 services, thus providing a foundation for designers of future Web system access, applications, and service, including those working on *wise*Pad (see Figure 3).

Many component technologies for *wise*Pad services, most notably captioning, are market-ready today; if combined, they would be close to being a breakthrough for improving the quality of life for many mobile vision-, hearing-, and/or speech-impaired users. Easy-to-use applications for all wearable computer users, not only the handicapped, are emerging. Heartwarming and worthy of celebration, ubiquitous services, including *wise*Pad, would contribute to personal empowerment, contribution, and inclusion for all.

_wise_Pad user services.

Composite *wise*Pad Services	User Functionality
Tell-IT	Allows gesture-based commands and communications among devices
VoiceIT	Converts user-targeted text to speech
CaptionIT	Captions audio from various feeds
Caption-and-PictureIT	Captions audio and embeds pictographs in resulting text
PictureIT	Embeds pictographs in text
SimplifyIT	Rewrites complex text and presents the user with simple text
SummarizeIT	Summarizes targeted text
Simplify-and-PictureIT	Simplifies and embeds pictographs in text
SimSumandPICIT	Gives users a simple summary illustrated with pictographs
ChangeIT	Transforms multimedia in user-defined ways, including zoom images, magnify parts of Web pages, color graphs, and change contrasts in presented materials
TransformIT-to-another-language	Provides translation of audio and text to other user-specified languages

References

1. Basson, S.H. and Kanevsky, D. *Universal Closed-Caption Portable Receiver, U.S. Patent Application 7,221,405*, May 22, 2007. IBM, Armonk, NY.
2. Fein, R.A., Dolan, W.B., Messerly, J., Fries, E.J., Thorpe, C.A., and Cokus, S.J. *Document Summarizer for Word Processors, U.S. Patent 5,924,108*, July 13, 1999. Microsoft Corp., Redmond, WA.
3. Haritaoglu, I. InfoScope: Link from real world to digital information space. In *Proceedings of the International Conference on Ubiquitous Computing* (Atlanta, GA, Sept. 30). Lecture Notes in Computer Science Series 2201, Springer, Berlin, 2001, 247–255.
4. Jutla, D.N. and Kanevsky, D. *What's Next Charles Schwab? User Space in Personal Investing*. IBM Research Report. IBM Research Division, Nov. 6, 2006; domino.research.ibm.com/library/cyberdig.nsf/papers/F3568D6DB7557C0685257222006F47A5/$File/rc24099.pdf.
5. Kanevsky, D. and Zlatsin, A. *Controlling Devices' Behaviors Via Changes in their Relative Locations and Positions, U.S. Patent 20080174547*, July 24, 2008. IBM, Armonk, NY; appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearch-bool.html&r=12&f=G&l=50&co1=AND&d=PG01&s1=kanevsky.IN.&OS=IN/kanevsky&RS=IN/kanevsky.
6. Kanevsky, D. and Zadrozny, W.W. *Virtual Shadow Briefcase in Servers Supporting Moving Embedded Clients, U.S. Patent No 6,912,580 B1*, June 28, 2005. IBM, Armonk, NY; www.freepatentsonline.com/6912580.html.
7. Kanevsky, D. and Sorenson, J.S. *Head-Mounted Display Content Transformer. U.S. Patent 6,738,535*, May 18, 2004. IBM, Armonk, NY; www.freepatentsonline.com/6738535.html.
8. Microsoft and Forrester Research. *The Market for Accessible Technology: The Wide Range of Abilities and Its Impact on Computer Technology.* Research Report. Forrester Research, Cambridge, MA, 2003; www.microsoft.com/enable/research/phase1.aspx.
9. Toossi, M. A century of change: U.S. labor force 1950–2050. *Monthly Labor Review 125*, 5 (May 2002), 15–28.

Dawn N. Jutla (dawn.jutla@smu.ca) is a full professor and chair of the Department of Finance, Information Systems, and Management Science in the Sobey School of Business at Saint Mary's University, Nova Scotia, Canada.

Dimitri Kanevsky (kanevsky@us.ibm.com) is a researcher, master inventor, and project manager in the Speech and Language Algorithms Department in the IBM T.J. Watson Research Center, Yorktown Heights, NY.

DOI:10.1145/1435417.1435435

Economic and social sciences will drive Internet protocols and services into the future.

BY JOAN FEIGENBAUM, DAVID C. PARKES, AND DAVID M. PENNOCK

Computational Challenges in E-Commerce

COMPANIES AND INDIVIDUALS are using computer networks to conduct increasing amounts of their daily business. Web search engines auctioned some $10 billion of ad space in 2007, accounting for almost half of all online advertising revenue. Sales at Amazon.com were $4.13 billion in the first quarter of 2008, including a fast-growing revenue stream from selling Web services to other e-commerce companies. At eBay, sales reached $15.7 billion in the second quarter, with 84.5 million active users.

This explosion of large-scale e-commerce poses new computational challenges that stem from the need to understand incentives. Because individuals and organizations that own and operate networked computers and systems are autonomous, they will generally act to maximize their own self-interest—a notion that is absent from traditional algorithm design. In this article, we provide an overview of four areas of computation in which incentives play a crucial role: resource allocation, knowledge integration, peer production and interaction, and security and privacy.

Resource Allocation

Allocating scarce resources—from bread to bytes—is a fundamental process that permeates economics and, indeed, society. Participants declare their perceived value for the resource and the market computes the best (for example, value-maximizing) allocation and the prices that participants should pay.

One decentralized prescription for resource allocation is an auction. Classical auctions emphasize simple rules for setting allocations and prices, which can be determined manually. Many of the largest marketplaces in the world, including financial exchanges, are at their core based on these centuries-old procedures. In one week of March 2008, the U.S. treasury sold, largely through manual means, more than $22 billion in three-month treasury bills.

Modern computer systems can support much richer and more flexible mechanisms. Governments use auctions to allocate property rights such as wireless spectrum (with worldwide proceeds exceeding $100 billion by the end of 2001). *Combinatorial* auctions allow bidders to express values for bundles of goods—for example, in assigning a higher value to two adjacent properties than the sum of the values assigned to each.[12] Generalized combinatorial auctions with rich and natural forms of expressiveness—volume discounts, side constraints, and bundling requirements, among others—are used to determine billions of dollars of spending within the supply chain, even though the problem is NP-hard. For example, they are used to source truckload-transportation logistics for Procter & Gamble, Walmart, and Target.[30]

Advertising is a business based on allocating *attention*, one of the scarcest and most valuable of resources. Media companies capture attention by providing information or entertainment and typically sell a fraction of that attention to advertisers.

Historically, advertising sales featured straightforward allocation rules and manual negotiations. But now more aspects of advertising, including its sale, delivery, and measurement, are being automated. Web-search engines such as Google and Yahoo! have led the way, selling space beside particular search queries in continuous dynamic auctions worth billions of dollars annually.

Auctions and exchanges for all types of online advertising—including banner and video ads—are commonplace at present, and they are run by startups and Internet giants alike. Advertisers can buy not only space but also contextual events—such as clicks from a specific user on a specific property at a specific time—or, more generally, bundles of contextual events. An ecosystem of third-party agencies has grown to help marketers manage their increasingly complex ad campaigns.

The rapid emergence of new modes for selling and delivering ads is fertile ground for research, both from economic and computational perspectives.[25] Edelman et al.[15] and Varian[31] model how advertisers bid in search-ad auctions. Essentially, the advertisers raise their bids until they reach a point of indifference between staying where they are and swapping with the advertiser above them on the page. The authors show that this bidding strategy forms the basis of a symmetric Nash equilibrium and, in a nice example where theory aligns with practice, that real bidding behavior is largely consistent with the model.

A number of questions drive research in ad auctions and exchanges. What mechanisms increase advertiser value or publisher revenue? What user and content attributes contribute to variation in advertiser value? How can bids for different contingencies (impressions, clicks, or conversions) be integrated and optimized over time? What constraints on supply and budget make sense? How should advertisers and publishers bid? How can publishers and advertisers incorporate learning and optimization (while trying to balance exploration and exploitation)? How do practical constraints such as real-time delivery affect design? How is automation changing the advertising industry? More information can be found in the *Proceedings of the Workshop on Ad Auctions* series.[33]

Knowledge Integration

The eliciting and aggregation of information from diverse and frequently self-interested sources is in general called "knowledge integration," with a particular case being "price discovery"—a side effect of market-based resource allocation. The balance point of supply and demand reveals the negotiated value of the resource.

In some cases, the value revealed in prices can rival or eclipse the value of trade. For example, the price of an asset that pays $1 if a category-5 hurricane hits Florida in 2009 can be seen as a probabilistic forecast of this catastrophic event. The value of an actual and more accurate forecast could run into the millions of dollars.

A "prediction market" is a market designed primarily for price discovery rather than resource allocation, and this alternate focus leads to a different prioritization of design goals. For example, the market operator may be happy to pay for the information it seeks, instead of enforcing neutral or positive revenue. Trading is not the end goal but a means to the end of acquiring complete, accurate, and timely information. For example, the Iowa Electronic Market forecasts the outcomes of political elections, and intrade.com predicts events ranging from the outbreak of avian flu to Osama bin Laden's capture.

Liquidity and expressiveness play important roles in prediction markets. If a trader with information cannot reveal it to the market, either because illiquidity prevents matching with another trader or because the market does not support the way in which the trader wants to express information, then the mechanism may fail.

Designing prediction markets to improve liquidity and expressiveness poses substantial though not insuperable computational challenges.[25] Liquidity can be addressed through the use of automated market makers that are always willing to buy and sell at some prices and that adjust prices dynamically to ensure a bound on their worst-case loss.[18, 26] Expressiveness is gained at an often-severe computational cost,

thus placing a difficult computational problem in the lap of the auctioneer striving to bring matched traders together or of the market maker trying to (implicitly) maintain an exponential number of prices.[10] In some instances, there is a reasonable and useful compromise between expressiveness and computational complexity.[11]

A more direct means of obtaining information is to pay an expert, though payment of a fixed amount does not motivate the expert to be either truthful or careful, let alone to actively seek out new information. A "scoring rule" is a payment function that depends on the expert's prediction and the actual outcome in such a way as to motivate truthful participation.[32] Shared scoring rules can form the basis of self-financing (budget-balanced) wagering mechanisms to obtain multiple individual forecasts.[23] Indeed, the line between scoring rules and markets becomes blurred. For example, the most common automated market maker used for prediction markets can be viewed as a sequential shared scoring rule.[18]

Rating systems are forums for gathering subjective opinions on a variety of things, such as movies, restaurants, or trading partners. Unlike forecasts, rating and reputation systems have no fundamental truths on which to base rewards. Although rating systems may provide personalized recommendations or advice, the incentives of raters may not align with these goals. Real rating systems do provide considerable value, despite the persistence of spam and pollution. Designing rating systems that are resistant to manipulation is an important challenge, requiring both good algorithms and good economics.[28]

Peer Production and Interaction

"Peer production," a term coined by Benkler,[6] refers to large-scale collaboration that is not based on price signals and occurs outside of the typical hierarchies and reward structures provided by firms. Salient examples of successful artifacts of peer production include Wikipedia and Linux. *Social* production, a more general phenomenon, describes the output of social relations—for example, the videos that people upload to YouTube and the content on social-networking sites such as Facebook. Taken together, these activities make up an energetic swath of the e-commerce landscape, both because of the opportunities to promote nontraditional production through appropriate feedback, trust, and accounting mechanisms and because of opportunities for targeted advertising and monetization efforts.

How can it be that peer production seems to succeed despite the widely accepted economic model of people as selfish and rational actors? Nontraditional motivations, such as hedonic pleasure from doing useful work and civic pride in observing community and societal norms, seem to be a part of the story. Indeed, the established field of behavioral economics seeks to explain people's social or "other-regarding" preferences.[16] Monetary rewards can actually lead to the crowding out of social motivations and to increasingly selfish behavior. In one noted example, when a country moved from a system of voluntary blood donations to one with small payments it found that donation rates went down instead of up.[17]

Despite positive examples of peer production, there is little formal knowledge about the design of successful peer-production systems. Pertinent methodologies seem necessarily more indirect than those of economic mechanism design,[20] given that actions are intrinsically voluntary. One challenge is to design systems that observe behaviors with a view to learning (social) preferences. Can environments be usefully modulated through appropriate constraints and affordances (for example, with moderator rights and other tiers), the assignment of rewards (say, gold stars),[35] and the ability to aggregate and disseminate information about peers (such as through scoreboards and social context)?

Lessons learned from peer-to-peer file sharing suggest that incentive considerations will remain important even in the presence of other-regarding preferences. Early protocols failed to provide appropriate incentives for the uploading of files, and systems such as Gnutella suffered from a large amount of free-riding.[3] The BitTorrent protocol addressed this problem by limiting a user's download rate according to upload history, thus mitigating incentives to free-ride. But such tit-for-tat during bilateral peering introduces its own market inefficiencies; for example, users cannot contribute upload resources in return for credit for later downloads unless centralized trackers are employed. Hybrid systems that allow for accounting and some form of currency may be of interest, though they have associated challenges with regard to dynamic stability.[22] There are interesting directions as well in the introduction of social structure into peering societies.[27]

Reputation and trust metrics have an important role to play in Internet commerce,[29] yet finding the right design can be quite a subtle problem. For example, one study suggests that good reputations of sellers on eBay are sometimes the result of a badly designed feedback protocol. A seller, who has the "last move," can punish a buyer who leaves negative feedback; the seller may respond with negative feedback of his own about the buyer.[8]

Recent work has formalized the challenges of providing provably non-manipulable trust metrics in graph-theoretic terms. Suppose that players are nodes and that player j can choose to lay down an edge (j, k) to another player k (possibly weighted), indicating a trust relationship; and suppose further that nodes can misrepresent trust information and create new ("fake") nodes. Various algorithms can be defined to compute pairwise trust between nodes, and their informativeness and manipulability can be compared. For instance, the EigenTrust algorithm[21] is vulnerable to Sybil attacks; one player can lay down multiple fake nodes that pump reputation flow in its direction. Other algorithms are more robust. Consider, for instance, defining pairwise trust i-j (i's trust of j) as the number of hops on the shortest path from i to j in an unweighted directed graph. Player j cannot reduce the i-j path length and improve the i-j trust by adding fake nodes and (directed) edges, as these nodes and edges can only affect shortest paths that flow *through* j and thus must leave the i-j path unaffected.[4] A similar argument establishes that node j cannot reduce the i-k trust for any node k that i trusts *more* than j. The only paths affected would be those that go through j and therefore ultimately reach nodes less trusted by i than j.

One outstanding challenge in the area of trust metrics is to find a satisfactory definition of informativeness; current axiomatic approaches appear unsatisfactory in this regard. With such a definition in hand, tensions between robustness and informativeness could be explored and perhaps also allow for mitigating factors such as the presence of other-regarding and altruistic actors within peering systems.

Security and Privacy

Research in security and privacy has been a central theme in academic computer science for more than 30 years. But although many clever algorithms, protocols, and devices have been rigorously analyzed, experimentally deployed, and even commercially developed, few of these solutions are in widespread use.

In 2001, Anderson initiated a new security-research direction by questioning the tacit assumption that security is primarily a technical problem. “[I]nformation insecurity is at least as much due to perverse incentives” as to technological failure, he said.“ Many of the problems can be explained more clearly and convincingly using the language of microeconomics: network externalities, asymmetric information, moral hazard, adverse selection, liability dumping, and the tragedy of the commons.”[5] In the years since Anderson’s seminal paper appeared, most aspects of security and privacy research have been suffused with economic considerations. We confine ourselves here to a brief discussion of the role of economics in user privacy, spam, and digital content distribution. More information can be found, for example, at the Web site of the Workshop on Economics of Information Security.[34]

That the increase in e-commerce has coincided with an erosion of consumer privacy seems paradoxical. After all, the ability to shop without having to stand face-to-face with others in a store seems privacy enhancing. Encryption, pseudonyms, electronic cash, and numerous other techniques purport to enable online consumers to protect their personal information and their identities. In practice, however, the trend has been for online merchants to ask for more information about their customers and for customers, even sophisticated ones who claim to value privacy, to provide this information when asked. Acquisti[1] offers the following explanation: “Behind a privacy intrusion there is often an economic trade-off. ... [I]ndividuals want to avoid the misuse of the information they pass along to others, but they also want to share enough information to achieve satisfactory interactions; organizations want to know more about the parties with which they interact, but they do not want to alienate them with policies deemed as intrusive.”

That the increase in e-commerce has coincided with an erosion of consumer privacy seems paradoxical. After all, the ability to shop without having to stand face-to-face with others in a store seems privacy enhancing.

Behavioral economics combined with experimental psychology yields the following general explanation for privacy erosion. Even “sophisticated” consumers who value privacy will often compromise it to improve their position in an ongoing transaction. They know that loss of control over their private information poses a long-term threat, but they cannot assess the long-term risk accurately enough to compare it to the short-term gain in the ongoing transaction.[2]

Although not strictly an “e-commerce” issue, privacy in public databases[9] is relevant to this discussion. In publicly available, sanitized aggregations of sensitive data about individuals, the canonical example of which is a census database, there is a natural tension between *utility* of the collection and *privacy* of the individuals, and communities often opt for utility.

Among all forms of unwanted communication, email spam has received the most attention. But the phenomenon also includes link spam (which degrades search quality), shilling (which degrades reviewer and recommender systems), and click fraud. Given this rogue’s gallery, there is widespread agreement that the incentive structure of next-generation network services must be designed more carefully.[13] Techniques for incentive-based prevention of unwanted communication include pricing via processing,[14] attention bonds,[24] and click-based learning algorithms.[19]

Nowhere is the inadequacy of a purely technical approach to information security clearer than in the realm of copyright enforcement. Digital-content distributors do have some control over the prices of their products and the flexibility with which purchased products can be used (for example, the extent to which they can be copied, shared, or modified), but although more flexible products may fetch higher prices they may also show dampened sales. Taking into account the fact that some digital environments are highly permeable (for example, they have higher rates of social contact or network bandwidth), Bergemann et al.[7] show the following:

- If users are homogenous, there is a critical permeability level up to which all users buy the product, and the optimal price and flexibility levels decrease as permeability increases. When permeability rises above the critical value, the number of users who buy decreases, and the optimal price and flexibility stay constant.

- A platform vendor who also sells digital content (which he licenses from copyright owners) will find it optimal to allow very flexible use by platform customers and to set a low price for content; most of the vendor's profit is from platform sales. Copyright owners' revenue is low, because prices are low and sharing of content is common.
- Tension between the platform vendor and copyright owner is observed in the real world. In 2005, the *Financial Times* quoted a music executive as saying, "Our music is not something to be given away to sell iPods."

Conclusion

Nearly all mass-market services—email, Web search, social networks, recommendations, user content, and ad networks, for example—seem beset by attempts to manipulate and distort. New services, such as the semantic Web, are assumed to be immune only while they remain niche. Yet to date the design of Internet protocols and services, including monetization efforts, have often been guided by technology rather than economics; and users have been modeled in caricature as either cooperative or malicious, the later to be dealt with by security measures. In truth, few users are benevolent and fewer still are vandals. Even the most vilified of participants—spammers—are acting in response to rational economic considerations. Understanding how to align the incentives of participants with systemwide objectives is fundamental to the design of the next generation of Web-scale services. Increasingly, design teams will require dual expertise in social science and computer science, adding competence in economics, sociology, and psychology to more traditionally recognized requirements such as algorithms, interfaces, systems, machine learning, and optimization.

In this article we focused on computational challenges in Internet-based (specifically Web-based) commerce. Yet people are starting to use smartphones and other handheld devices to do much of what they used to do with Internet-connected desktop or laptop PCs. To what extent these handheld devices will ultimately behave like networked PCs is unclear, and thus we leave to a future article the needed discussion of computational challenges in mobile commerce.

Acknowledgments

Joan Feigenbaum was supported in part by NSF grants CNS-0331548, CNS-0428422, and IIS-0534052. David Parkes was supported in part by NSF grants DMS-0631636, IIS-0534620, IIS-0238147, an Alfred P. Sloan Research Fellowship, and Yahoo! Research and Microsoft Research awards; he gratefully acknowledges helpful conversations with Yochai Benkler, Chris Kelty, Johan Pouwelse, and Sven Seuken. C

References

1. Acquisti, A. *The Economics of Privacy*; www.heinz.cmu.edu/~acquisti/economics-privacy.htm.
2. Acquisti, A. Privacy in electronic commerce and the economics of immediate gratification. In *Proceedings of the 5th Electronic Commerce Conference*. ACM (2004), 21–29; www.heinz.cmu.edu/~acquisti/papers/privacy-gratification.pdf.
3. Adar, E. and Huberman, B. Free riding on Gnutella. *First Monday 5* (2000), 2.
4. Altman, A. and Tennenholtz, M. An axiomatic approach to personalized ranking systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (2007), 1187–1192.
5. Anderson, R. Why information security is hard—An economic perspective. In *Proceedings of the 17th Annual Computer Security Applications Conference* (2001), 358–365; www.cl.cam.ac.uk/~rja14/Papers/econ.pdf.
6. Benkler, Y. Coase's penguin, or, Linux and the nature of the firm. *The Yale Law Journal 112* (2002), 369–446.
7. Bergemann, D., Eisenbach, T., Feigenbaum, J., and Shenker, S. Flexibility as an instrument in digital rights management. In *Proceedings of the 2005 Workshop on Economics of Information Security*; http://infosecon.net/workshop/pdf/50.pdf
8. Bolton, G., Greiner, B., and Ockenfels, A. Engineering trust: Strategic behavior and the production of reputation information. Working paper, Harvard Business School, 2007.
9. Chawla, S., Dwork, C., McSherry, F., Smith, A., and Wee, H. Toward privacy in public databases. In *Proceedings of the 2nd Theoretical Cryptography Conference*. Springer LNCS 3378 (2005), 363–385; http://pages.cs.wisc.edu/~shuchi/papers/privacy-TCC.pdf
10. Chen, Y., Fortnow, L., Lambert, N., Pennock, D.M., and Wortman, J. Complexity of combinatorial market makers. In *Proceedings of the 9th Conference on Electronic Commerce. ACM* (2008), 190–199.
11. Chen, Y. Goel, S., and Pennock, D.M. Pricing combinatorial markets for tournaments. In *Proceedings of the 40th Symposium on Theory of Computation* (2008), 305–314.
12. Cramton, P., Shoham, Y., and Steinberg, R., Eds. *Combinatorial Auctions*. MIT Press, Cambridge, MA, 2006.
13. Dhajima, R., Tygar, D., and Heart, M. Why phishing works. In *Proceedings of the Conference on Human Factors in Computer Systems*. ACM (2006), 581–590.
14. Dwork, C. and Naor, M. Pricing via processing or combating junk mail. *Crypto '92*. Springer LNCS 740 (1992) 139–147; www.wisdom.weizmann.ac.il/~naor/PAPERS/pvp.ps.
15. Edelman, B., Ostrovsky, M., and Schwarz, M. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review 97* (2007), 242–259.
16. Fehr, E. and Camerer, C. Measuring social norms and preferences using experimental games: A guide for social scientists. In *Foundations of Human Sociality*, Henrich, J., Boyd, R., Bowles, S., Camerer, C., Fehr, E., and Gintis, H. Eds. Oxford University Press, Oxford, U.K., 2004.
17. Frey, B.S. and Jegen, R. Motivation crowding theory. *Journal of Economic Surveys 15* (2001), 589–611.
18. Hanson, R. Logarithmic market scoring rules for modular combinatorial information aggregation. *Journal of Prediction Markets 1* (2007), 3–15.
19. Immorlica, N., Jain, K., Mahdian, M., and Talwar, K. Click fraud resistant methods for learning click-through rates. In *Proceedings of the 1st Workshop on Internet and Network Economics 3828*. Springer LNCS (2005) 34–45; www.ece.northwestern.edu/~nickle/pubs/clickfraud.pdf.
20. Jackson, M.O. Mechanism theory. *Encyclopedia of Life Support Systems*. Derigs, U., Ed. EOLSS Publishers, Oxford, U.K., 2003.
21. Kamvar, S.D., Schlosser, M.T., and Garcia-Molina, H. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International World Wide Web Conference*, (2003), 640–651; http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.3109.
22. Kash, I.A., Friedman, E.J., and Halpern, J.Y. Optimizing scrip systems: Efficiency, crashes, boarders, and altruists. In *Proceedings of the 8th Conference on Electronic Commerce*. ACM (2007), 305–315.
23. Lambert, N., Langford, J., Wortman, J., Chen, Y., Reeves, D., Shoham, Y., and Pennock, D.M. Self-financed wagering mechanisms for forecasting. In *Proceedings of the 9th Conference on Electronic Commerce*. ACM (2008), 170–179.
24. Loder, T., Van Alstyne, M., and Wash, R. An economic response to unsolicited communication. *Advances in Economic Analysis and Policy 1* (2006), Article 2; www.bepress.com/bejeap/advances/vol6/iss1/art2/.
25. Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V.V., Eds. *Algorithmic Game Theory*. Cambridge University Press, 2007.
26. Pennock, D.M. A dynamic pari-mutuel market for hedging, wagering, and information aggregation. In *Proceedings of the 5th Conference on Electronic Commerce*. ACM, 2004, 170–179.
27. Pouwelse, J.A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D.H.J., Reinders, M.J.T, van Steen, M.R., and Sips, H.J. TRIBLER: A social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience 20* (2008), 127–138.
28. Resnick, P. and Sami, R. The influence-limiter: Provably manipulation-resistant recommender systems. *Recommender Systems Conference*. ACM, 2007, 25–32.
29. Resnick, P., Zeckhauser, R., Swanson, J., and Lockwood, K. The value of reputation on eBay: A controlled experiment. *Experimental Economics 9* (2006), 79–101.
30. Sandholm, T. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine 28* (2007), 45–58.
31. Varian, H.R. Position auctions. *International Journal of Industrial Organization 25* (2007), 1163–1178.
32. Winkler, R., Munoz, J., Cervera, J., Bernardo, J., Blattenberger, G., Kadane, J., Lindley, D., Murphy, A., Oliver, R., and Rios-Insua, D. Scoring rules and the evaluation of probabilities. *TEST 5* (1996), 1–60.
33. Workshops on Ad Auctions; http://research.yahoo.com/workshops/ad-auctions-2008/pastworkshops.html.
34. Workshops on Economics of Information Security; http://weis2008.econinfosec.org/.
35. Zhang, H. and Parkes, D.C. Value-based policy teaching with active indirect elicitation. In *Proceedings of the 23rd Conference on Artificial Intelligence*. (2008), 208–214.

Joan Feigenbaum (joan.feigenbaum@yale.edu) is the Grace Murray Hopper Professor of Computer Science at Yale University, New Haven, CT.

David C. Parkes (parkes@eecs.harvard.edu) is Gordon McKay Professor of Computer Science at Harvard University, Cambridge, MA.

David M. Pennock (pennockd@yahoo-inc.com) is a principal research scientist at Yahoo! Research, New York, NY.

DOI:10.1145/1435417.1435436

Technical Perspective
Customizing Media to Displays

By Harry Shum

A FEW YEARS AGO, I bought a widescreen TV with an aspect ratio of 16:9. It is great for watching movies shot with a wide-screen format. However, on most other occasions, I am faced with a dilemma: If I choose the option to fill the entire screen, everything looks wider than normal, while preserving the aspect ratio of the video means seeing wasted space at both ends.

There is a mind-boggling array of displays that are readily available, from large plasma displays and high-resolution LCDs to low-resolution cellphone screens. These displays differ greatly in resolutions and aspect ratios. The problem is, images and videos are captured at fixed resolutions and aspect ratios, and from personal experience, viewing them properly in a display can be a challenge.

What, then, is the correct way of displaying media? Global scaling solves part of the problem, but naively stretching or squashing one of the dimensions to fill the screen introduces undesirable distortions. Cropping is not a satisfactory solution either because important elements in the scene may either be partially removed or totally cut out. We need a solution that intelligently *customizes media to displays*.

The answer may well lie in the work of Ariel Shamir and Shai Avidan. Their technique, intriguingly called "seam carving," cuts out or adds pixels to swaths of areas deemed less important. The importance can be measured by contrast or need to preserve humans or objects. Given the energy function that measures this importance, the process of removing pixels to minimize this energy function is nontrivial. This is because we must preserve both the rectangular shape and visual coherence of the image. Shamir and Avidan devised a simple but powerful idea: carve (remove) seams iteratively.

A seam is a connected path of low-energy pixels crossing the image from top to bottom or from left to right. Their seam carving algorithm changes the aspect ratio of the image by iteratively carving the seams with the lowest importance, horizontally or vertically. The optimal seam at each iteration can be found using dynamic programming.

Herein lies the magic of seam carving: removing a seam has only a local impact and the produced visual artifacts are globally imperceptible. As a result, seam carving maintains both a rectangular shape and visual coherence of the image.

To enlarge the image, the seam carving process is run in reverse—by adding interpolated pixels along the lowest energy seam. The authors have also demonstrated other applications of seam carving, such as content amplification, object removal, multisize image format, and last but not least, video resizing. Video resizing is nontrivial because of the need for temporal coherence in addition to spatial coherence. Shamir and Avidan cleverly achieved video resizing by casting the problem as a 3D graph with 2D manifold (instead of 2D graph with 1D curve for images).

We need a solution like the one Shamir and Avidan explore here. However, there are two important issues that must be addressed before such a solution is exposed to the masses. First, there must be real-time performance (that is, real-time rendering of media). Even if the algorithm is highly optimized, I would imagine it is difficult to achieve real-time resizing for high-resolution images and HD videos. Shamir and Avidan recommend precomputing the resizing operations for the most popular resolutions and aspect ratios and storing the vertical and horizontal seam index maps. The player or TV set recognizes the display format, fetches such relevant metadata information, and re-renders the original video appropriately. While this is a good idea, in order for this solution to be practical, there needs to be an efficient compression scheme for the seam index maps, especially for video. The other issue is related to algorithmic robustness: How can the intent, tenor, and attractiveness of media be preserved after it has been resized? Can these qualities be reliably codified? Human visual attention has been modeled to some extent (see, for example, the work of Itti, Koch, and Niebur[1]), but is such a model enough?

Independent of these questions, as any computer vision scientist will tell you, completely automatic vision techniques are typically not foolproof. All techniques come with assumptions that may not be satisfied all the time. Consumers may not be forgiving if their video looks less than attractive—a horde of people brandishing pitch forks and torches come to mind. Like it or not, I believe we will need a human in the loop for media customization. I am a big proponent of *interactive computer vision*, that is, the concept of judiciously adding interaction to complement what can be automated. This is acceptable in the context of media customization because it needs only to be done once for each video. (Plus, it may spawn a sizeable cottage industry.) The trick, then, is to design an interface that minimizes manual input. Shamir and Avidan's innovative algorithm should be adapted to take into consideration manual annotation to preserve the intent of the media.

I now look at my wide-screen TV and wistfully think, if only it is possible to customize media to displays *now*...

Reference

1. Itti, L., Koch, C., and Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (Nov. 1998), 1254–1259

Harry Shum (hshum@microsoft.com) is a Fellow of ACM and a Corporate Vice President of Microsoft Corporation, Redmond, WA.

DOI:10.1145/1435417.1435437

Seam Carving for Media Retargeting

By Ariel Shamir and Shai Avidan

Abstract

Traditional image resizing techniques are oblivious to the content of the image when changing its width or height. In contrast, media (i.e., image and video) retargeting takes content into account. For example, one would like to change the aspect ratio of a video without making human figures look too fat or too skinny, or change the size of an image by automatically removing "unnecessary" portions while keeping the "important" features intact. We propose a simple operator; we term *seam carving* to support image and video retargeting. A seam is an optimal 1D path of pixels in an image, or a 2D manifold in a video cube, going from top to bottom, or left to right. Optimality is defined by minimizing an energy function that assigns costs to pixels. We show that computing a seam reduces to a dynamic programming problem for images and a graph min-cut search for video. We demonstrate that several image and video operations, such as aspect ratio correction, size change, and object removal, can be recast as a successive operation of the seam carving operator.

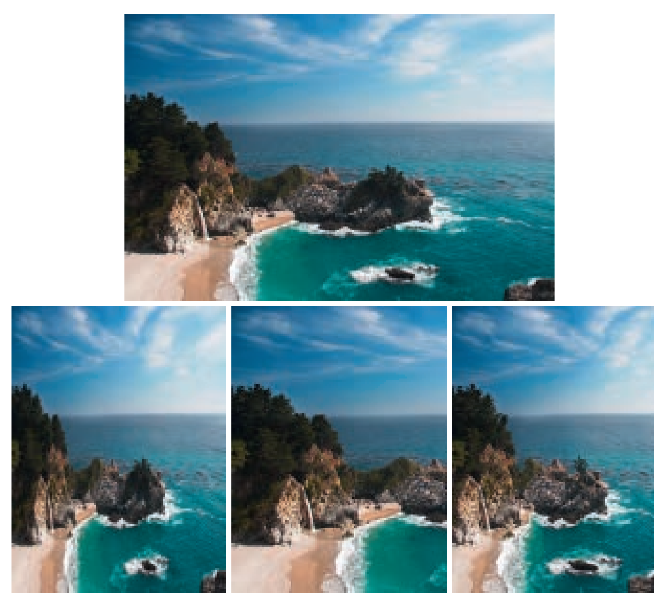
1. INTRODUCTION

The diversity and versatility of display devices today imposes new demands on digital media. Designers must create different alternatives for web-content and design different layouts, possibly even dynamic layouts, for different devices. Nevertheless, images and video, although being key elements in digital media, typically remain rigid in size and shape and cannot deform to fit different displays. To accommodate high definition TVs, computer screens, PDAs, and cell phones not only the resolution, but often the aspect ratio of the media must be adjusted. Standard image scaling is not sufficient since it is oblivious to the image content and typically can be applied only uniformly. Cropping is limited since it can only remove pixels from the image periphery and cannot support expansion (Figure 1). More effective resizing can only be achieved by considering the image content and not only geometric constraints. We propose a simple operator, we term *seam carving*, which can change the size of images and video by gracefully carving out or inserting pixels in different parts of the media. Seam carving is based on the definition of an energy function defining the importance of pixels. In images, a seam is a *connected* path of low energy pixels crossing the image from top to bottom (or from left to right), and is *monotonic*, that is, including one, and only one, pixel in each row (or column). In video, a seam is a monotonic and connected low energy 2D manifolds passing through a 3D volume cube defined by stacking the video frames (Figure 2).

By successively removing or inserting seams we can reduce, as well as enlarge, the size of media in both directions. For size reduction, seam selection ensures that we remove more of the low energy pixels and fewer of the high energy ones from the source, while also taking care to insert the least amount of energy to the target. For enlarging, the order of seam insertion ensures a balance between the original media content and the artificially inserted pixels. These operators produce, in effect, a *content-aware* resizing of media.

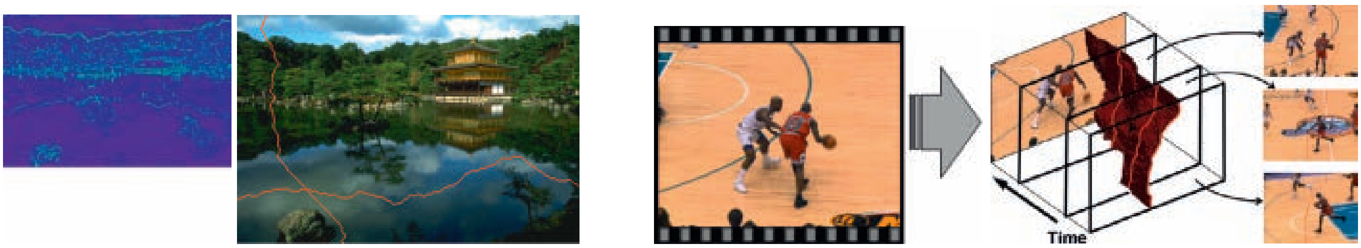
On images, the search for an optimal seam can be formulated using an efficient dynamic programming algorithm, which is linear in the number of pixels. For videos the dynamic programming approach is no longer applicable. We show how to define a graph such that running the min-cut/max-flow algorithm on it will create a cut that is a 2D

Figure 1: Content aware vs. content oblivious resizing: scaling (left) creates distortion, cropping (middle) removes important parts, while seam carving (right) preserves the image content.



This paper is a summary of two papers originally published in SIGGRAPH Proceedings: Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10; Improved seam carving for video retargeting. *ACM Trans. Graph. 27*, 3 (2008), 10.

Figure 2: A seam is a connected and monotonic path of low energy pixels (energy shown in small) connecting the two sides of the image either horizontally or vertically (shown in red). On video (right) the sequence of frames is combined to a 3D cube and a 2D monotonic and connected manifold is sought. The intersection of the manifold with each video frame defines the seams on the frame.



manifold. The intersection of this manifold with every video frame produces a valid seam on the frame. When applied to images this graph-cut formulation is equivalent to the one created by dynamic programming.

We illustrate the application of seam carving and insertion for aspect ratio change, media retargeting, and object removal. Furthermore, by storing the order of seam removal and insertion operations, we define *multi-size* images and video that support on-the-fly media retargeting at very high frame rates. Multisize media allows a designer to create retargeted media once and then let client applications retarget the media to various displays in real time.

Seam carving can support a whole range of image energy functions, while also reducing the created artifacts by looking forward at energy inserted into the resized media. To allow interactive control, we also provide a scribble-based user interface for adding or reducing the energy of a pixel or region and guide the desired results. This tool can also be used for object removal and for authoring multisize images.

2. BACKGROUND

Media resizing is a standard tool in many image and video editing tools. It works by uniformly resizing the media to a target size. Recently, there has been a growing interest in media retargeting that is driven largely by the growing number of mobile devices used to view digital content.[4, 12–14, 18, 22, 29] These seek to change the size of the image or video while maintaining the important features intact. The common approach for all media resizing works is first to define an importance map on the pixels of the media, and then use this map to guide some operator that reduces (or enlarges) the media size.

There are numerous ways to define importance in media. Some use top-down methods such as face detectors[23] or foreground-background segmentation,[20] while others use bottom-up methods that rely on visual saliency,[9] the detection of Regions Of Interest (ROI),[13, 18, 25] or motion.[29] Similarly, numerous operators are used to change the media size in a content-aware fashion. These include cropping,[7, 18] uniform scaling,[13] virtual camera motions,[7] and recently nonuniform warping.[8, 27, 29] More recently, some new approaches to media retargeting were introduced by Simakov[21] and Wei[28] where retargeting is formulated as the problem of patch similarity between source and target images. For a more thorough overview of previous work the reader is referred to Avidan[2] and Rubinsteing.[17]

In this work we present the novel seam carving operator for resizing. We mostly use image and video edge energies as the importance map, but can support any saliency map defined on pixels. We show results for reduction as well as expansion of images and video. We demonstrate object removal, and also define the notion of multisize media.

The use of seams for image editing and composition is prevalent.[1, 10, 16] However, none of these methods discuss the problem of image retargeting and none of them consider, as we do, computing a seam on a single image. Computing a seam can be done in a variety of ways, including Dijkstra's shortest path algorithm,[5] dynamic programming[6] that we use for images, or graph cuts[3] that we use for video.

3. THE OPERATOR

For ease of exposition, we will focus on images first, then we will show how to extend the seam carving operator to handle video. We start by presenting the operator in one direction allowing a user to change either the width or the height of an image. We then show how to optimally change both width and height of a given image. Next, we investigate the use of a different objective function that leads to visually more pleasing results. We follow with an extension to video retargeting based on a graph-cut formulation and finally describe *multisize media*, which is a new representation for images and video.

3.1. The one directional case

Our approach to content-aware resizing is to remove pixels in a judicious manner. Therefore, the question is how to choose the pixels to be removed? Intuitively, our goal is to remove unnoticeable pixels that blend with their surroundings. Since human vision is more sensitive to edges, we give high value to edges using the following simple energy function (see Figure 2, later, we elaborate on the definition of energy functions in Section 3.3):

$$e_1(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right| \tag{1}$$

Given this, or similar, energy function, assume we need to reduce the width of an image. One can think of several strategies to achieve this. For instance, an optimal strategy to preserve energy (i.e., keep pixels with high energy value) would be to remove the pixels with lowest energy in ascending order. This destroys the rectangular shape of the image, because we may remove a different number of pixels from each row (see Figure 3(e)). If we want to prevent the image from breaking we can remove an equal number of low energy pixels from every row. This preserves the rectangular shape of the image but destroys the image content by creating a zigzag effect (Figure 3(d)). To preserve both the shape and the visual coherence of the image we can use auto-cropping. That is, look for a subwindow, the size of the target image, that contains the highest energy (Figure 3(a)). Another possible strategy somewhat between removing pixels and cropping is to remove whole columns with the lowest energy. Still, artifacts might appear in the resulting image (Figure 3(b)). Therefore, we need a resizing operator that will be less restrictive than cropping or column removal, but can preserve the image content better than single pixel removals. This leads to our strategy of seam carving (Figure 3(c)).

Formally, let $\mathbf{I}$ be an $n \times m$ image and define an image *vertical seam* to be

$$\mathbf{s}^{x} = \{s_i^x\}_{i=1}^{n} = \{(x(i), i)\}_{i=1}^{n}, \text{ s.t. } \forall i, |x(i) - x(i-1)| \le 1 \quad (2)$$

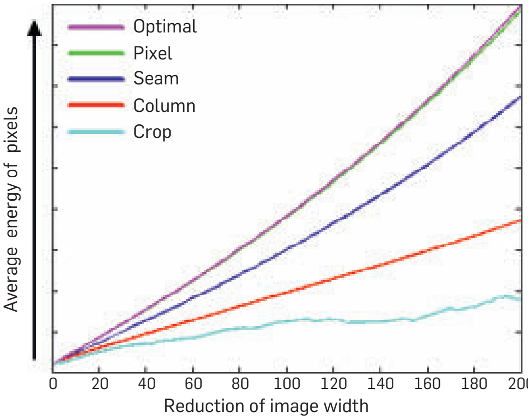
where x is a function x: $[1, \ldots, n] \rightarrow [1, \ldots, m]$ that is continuous in a discrete sense (two consecutive values of the function do not differ by more than 1). In other words, a vertical seam is a connected path of pixels in the image from top to bottom, containing one, and only one, pixel in each row of the image (see Figure 2). Similarly, if y is a discretely continuous function y: $[1, \ldots, m] \rightarrow [1, \ldots, n]$, then an image *horizontal seam* is

$$\mathbf{s}^{y} = \{s_j^y\}_{j=1}^{m} = \{(j, y(j))\}_{j=1}^{m}, \text{ s.t. } \forall j |y(j) - y(j-1)| \le 1 \quad (3)$$

The pixels of the path of seam $\mathbf{s}$ (e.g., vertical seam $\{s_i\}$) will therefore be $\mathbf{I}_s = \{\mathbf{I}(s_i)\}_{i=1}^{n} = \{\mathbf{I}(x(i), i)\}_{i=1}^{n}$. Note that similar to the removal of a row or column from an image, removing the pixels of a seam from an image has only a local effect: all the pixels of the image are shifted left (or up) to compensate for the missing path. The visual impact is noticeable only along the path of the seam, leaving the rest of the image intact.

Figure 3: Results of five different strategies for reducing the width of an image. (a) The original image and its e_1 energy function, (b) average energy graph, (c) best cropping, (d) removing columns with minimal energy, (e) seam removal, (f) removal of the pixel with the least amount of energy in each row, and finally, (g) global removal of pixels with the lowest energy, regardless of their position. The graph shows the energy preservation of each strategy.

(a) Original image and its energy



(b) Average energy

(c) Crop (d) Column (e) Seam (f) Pixel (g) Optimal

We define the cost of a seam as the sum of energy of its pixels $E(\mathbf{s}) = E(\mathbf{I}_\mathbf{s}) = \Sigma_{i=1}^{n} e(\mathbf{I}(s_i))$, and look for the optimal seam s^* that minimizes this cost:

$$s^* = \arg\min_{\mathbf{s}} E(\mathbf{s}) = \arg\min_{\mathbf{s}} \sum_{i=1}^{n} e(\mathbf{I}(s_i)) \qquad (4)$$

Although there seems to be an exponential number of possible seams, the optimal seam can in fact be found using dynamic programming in linear complexity. The first step is to traverse the image from the second row to the last row and compute the cumulative minimum energy M for all possible connected seams for each entry (i, j):

$$\begin{aligned} M(i,j) = e(i,j) + \min(&M(i-1,j-1), M(i-1,j), \\ &M(i-1,j+1)) \end{aligned} \qquad (5)$$

At the end of this process, the minimum value of the last row in M will indicate the end of the minimal connected vertical seam. Hence, in the second step we backtrack from this minimum entry on M to find the path of the optimal seam. The definition of M for horizontal seams is similar.

Reducing the width of an image by n pixels boils down to applying the seam carving operator n times. That is, in each time step we find the optimal seam in the image, remove the pixels associated with it and repeat the process for n times. It is worth noting that this approach resembles the dynamic shortest paths problem,[15] where finding the seam is equivalent to finding a shortest path on a graph, and removing the seam modifies the graph for the next iteration of shortest path search.

3.2. The two directional case

Image retargeting generalizes aspect ratio change by modifying the image size in both directions, such that an image $\mathbf{I}$ of size $n \times m$ can be retargeted to size $n' \times m'$. For the time being, we assume that $m' < m$ and $n' < n$. Using seam carving, this raises the question of what is the correct order of seam removal? Vertical seams first? Horizontal seams first? Alternate between the two? Or any other order? We define the search for the optimal order of seam removal as an optimization of the following objective function:

$$\min_{\mathbf{s}^x, \mathbf{s}^y, \delta} \sum_{i=1}^{k} E(\delta_i \mathbf{s}_i^x + (1-\delta_i)\mathbf{s}_i^y) \qquad (6)$$

where $k = r + c$, $r = (m - m')$, $c = (n - n')$ and δ_i is used as a parameter that determines if at step i we remove a horizontal or vertical seam: $\delta_i \in \{0, 1\}$, $\Sigma_{i=1}^{k}\delta_i = r$, $\Sigma_{i=1}^{k}(1-\delta_i) = c$.

We find the optimal order using a transport map $\mathbf{T}$ that specifies, for each desired target image size $n' \times m'$, the cost of the optimal sequence of horizontal and vertical seam removal operations. That is, entry $T(r, c)$ holds the minimal cost needed to obtain an image of size $n - r \times m - c$. We compute $\mathbf{T}$ using dynamic programming. Starting at $\mathbf{T}(0, 0) = 0$ we fill each entry (r, c) choosing the best of the two options—either removing a horizontal seam from an image of size $n - r \times m - c + 1$ or removing a vertical seam from an image of size $n - r + 1 \times m - c$:

$$\begin{aligned} \mathbf{T}(r,c) = \min(&\mathbf{T}(r-1,c) + E(\mathbf{s}^x(\mathbf{I}_{\mathbf{n-r-1\times m-c}})), \\ &\mathbf{T}(r,c-1) + E(\mathbf{s}^y(\mathbf{I}_{\mathbf{n-r\times m-c-1}}))) \end{aligned} \qquad (7)$$

where $\mathbf{I}_{\mathbf{n-r\times m-c}}$ denotes an image of size $n - r \times m - c$, $E(\mathbf{s}^x(\mathbf{I}))$ and $E(\mathbf{s}^y(\mathbf{I}))$ are the cost of the respective seam removal operation. We store a simple $n \times m$ 1-bit map which indicates which of the two options was chosen in each step of the dynamic programming. Choosing a left neighbor corresponds to a vertical seam removal while choosing the top neighbor corresponds to a horizontal seam removal. Given a target size $n' \times m'$ where $n' = n - r$ and $m' = m - c$, we backtrack from $\mathbf{T}(r, c)$ to $\mathbf{T}(0, 0)$ and apply the corresponding removal operations. Figure 4 shows an example of different retargeting strategies on an image.

3.3. Forward energy

To evaluate the effectiveness of the different strategies for content-aware resizing, we can examine the average energy of all the pixels in an image $\frac{1}{|\mathbf{I}|}\sum_{p\in I} e(p)$ during resizing. Randomly removing pixels should keep the average unchanged, but content-aware resizing should raise the average as it removes low energy pixels and keeps the high energy ones (Figure 3(f)).

Choosing to remove the seam with the least amount of energy from the image (Equation 5) may work for many

Figure 4: Optimal order retargeting: on the top is the original image and its transport map T. Given a target size, we follow the optimal path (white path on T) to obtain the retargeted image (top row, right). For comparison we show retargeting results by alternating between horizontal and vertical seam removal (top row, left), removing vertical seams first (bottom row, left), and removing horizontal seams first (bottom row, right).

images but ignores energy that is *inserted* into the target image. The inserted energy is due to new intensity edges created by previously nonadjacent pixels that become neighbors once the seam is removed (see, e.g., the steps artifacts in Figure 5). Assume we resize an image $I = I_{t=1}$ using k seam removals ($t = 1, ..., k$). To measure the real change in energy after a removal of a seam, we measure the difference in the energy of the image after the removal ($I_{t=i+1}$) and the energy of only those parts that were not removed in the previous image $I_{t=i}$ (i.e., the image energy $E(I_{t=i})$ minus the seam energy $E(S_i)$). The energy difference after the ith seam carving operation is

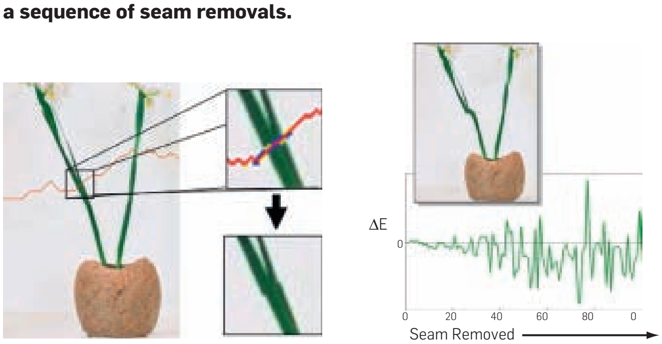
$$\Delta E_{t=i+1} = E(I_{t=i+1}) - [E(I_{t=i}) - E(S_i)] \tag{8}$$

As can be seen in Figure 5 (right), ΔE_t can actually increase as well as decrease for different seam removals using Equation 5. The figure also shows a specific example of a seam that inserts more energy to the image than it removes.

Following these observations, we formulate the *forward* looking criterion. At each step, we search for the seam whose removal inserts the minimal amount of energy into the image. These are seams that are not necessarily minimal in their energy cost, but will leave less artifacts in the target image, after removal. As the removal of a connected seam affects the image, and its energy, only at a local neighborhood, it suffices to examine a small local region near the removed pixel. We consider the energy introduced by removing a certain pixel to be the new "intensity-edges" created in the image. The cost of these intensity edges is measured as the differences between the values of the pixels that become new neighbors, after the seam is removed. Depending on the connectivity of the seam, three such cases are possible (see Figure 6). For each of the three possible cases, we define a cost respectively:

(a) $C_L(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j-1)|$
(b) $C_U(i,j) = |I(i,j+1) - I(i,j-1)|$
(c) $C_R(i,j) = |I(i,j+1) - I(i,j-1)| + |I(i-1,j) - I(i,j+1)|$

Figure 5: In some cases, choosing to remove the seam with the least amount of energy can create artifacts due to energy inserted into the image. Left: an example of the change in energy after a specific seam is removed. In some pixels energy is reduced (blue), while in others increased (yellow). Right: the actual change in energy ∅∆ for a sequence of seam removals.



We use these costs in a new forward-cumulative cost matrix MF to calculate the seams using dynamic programming. For vertical seams, each cost $MF(i,j)$ is updated using the following rule:

$$MF(i,j) = P(i,j) + \min \begin{cases} MF(i-1,j-1) + C_L(i,j) \\ MF(i-1,j) + C_U(i,j) \\ MF(i-1,j+1) + C_R(i,j) \end{cases} \tag{9}$$

where $P(i, j)$ is an additional pixel based energy measure, such as the result of high level tasks (e.g., face detector) or a user supplied weight, that can be used in addition to the forward energy cost. Figure 7 shows a comparison of the results using the two formulations.

3.4. Seam carving using graph cut

The dynamic programming formulation works well for images, as would a shortest path approach. However, neither one scales to video and hence we switch to a graph-cut formalism that can be used either for images or video. Graph partitioning and graph-based energy minimization techniques are widely used in image and video processing applications such as image restoration, image segmentation, object recognition, and shape reconstruction. A graph representing an image is created by connecting pixels based on their similarity together with some constraints. The graph is partitioned into disjoint subsets by removing, or cutting, some of its edges (arcs). For videos, it is often convenient to consider the sequence of frames as a 3D space–time cube,[11, 19, 24, 26] and use voxels connecting temporally instead of just pixels.

Our graph construction is a little different than most previous work both for images and for video. The challenge we face is to design a graph that produces only admissible cuts, for example, cuts whose intersection with each video frame (or image) will produce a valid seam, that is, it must satisfy two constraints:

Figure 6: Calculating the three possible vertical seam step costs for pixel $p_{i,j}$ using forward energy. After removing the seam, new neighbors (in gray) and new pixel edges (in red) are created. In each case the cost is defined by the forward difference in the newly created pixel edges. Note that the new edges created in row $i - 1$ were accounted for in the cost of the previous row pixel.

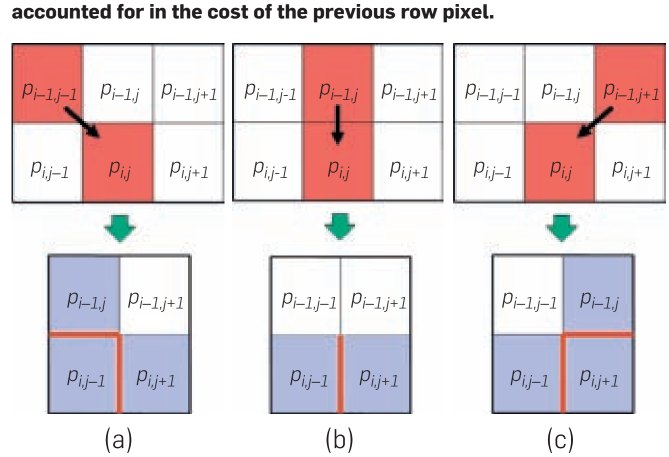
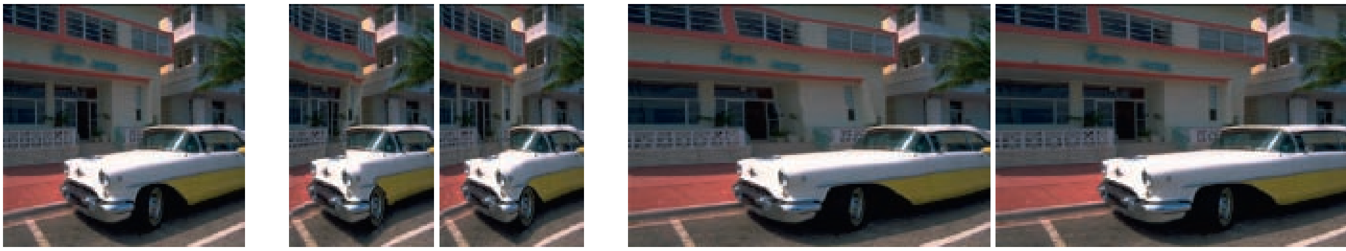


Figure 7: A comparison of results for reduction and expansion of the car image (leftmost) using least cost seam of Equation 5 (left in each pair) and least inserted cost seams of Equation 9 (right in each pair). We discuss image enlarging in Section 4.3.



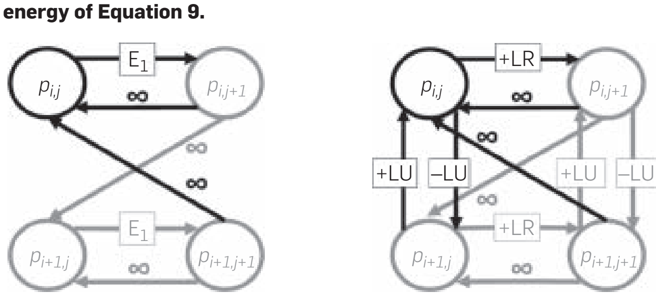
Monotonicity: the seam must include one, and only one pixel, in each row (or column for horizontal seams).
Connectivity: the pixels of the seams must be connected.

The monotonicity constraint ensures it is a function, while the connectivity constraint enforces continuity. Hence, the challenge is to construct a graph that guarantees the resulting cut will be a continuous function over the relevant domain. However, standard graph-cut-based constructions do not satisfy these constraints.

In our construction every node represents a pixel, and connects to its neighboring pixels in a grid-like structure. Virtual terminal nodes, S (source) and T (sink) are created and connected with infinite weight arcs only to the pixels of the leftmost and rightmost columns of the image, respectively (and only to the sides of the cube for video). Moreover, the key difference is that we use backward infinity arcs between the nodes. These arcs constrain the resulting cut. To define a seam from a cut, we consistently choose the pixels to the left of the cut arcs. The optimal seam is defined by the *minimum cut* which is the cut that has the minimum cost among all valid cuts. Figure 8 illustrates the construction of such graphs on images for the two approaches of Equations 5 and 9. These graphs constructions guarantee that the seam defined by the graph-cut algorithm would be the optimal, monotonic, and connected.

In the case of video we repeat the grid-like construction of the graph both in space (within a frame) and in time (between frames in one direction). In addition, we connect the source and sink nodes to the leftmost and rightmost pixels in all the frames of the video (for the case of vertical seams). The resulting cut of this 3D graph is a 2D manifold whose intersection with every video frame produces a valid seam (Figure 2, right). Hence, a vertical seam can be thought of as a discretely continuous function $S: Y \times T \rightarrow X$ from (row, time) to column. Note that similar constructions can be used to change the height as well as the *length* of the video in time. More details can be found in Rubinsteing.[17]

Figure 8: Two graph constructions that are used by the graph-cut algorithm illustrated by four neighboring nodes. The actual image graph is created by tiling these subgraphs across the image. The left graph is equivalent to the dynamic-programming seam carving approach of Equation 5 and right graph represents the new forward energy of Equation 9.



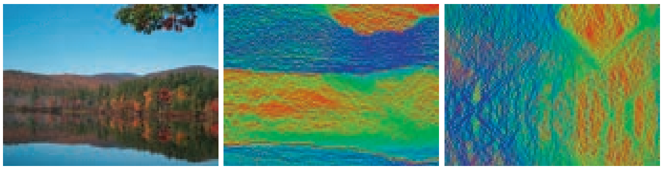
3.5. Multisize media

So far we have assumed that the user knows the target size ahead of time, but this might not be possible in all cases. Consider, for example, an image embedded in a web page. The web designer does not know, ahead of time, at what resolution the page will be displayed and therefore cannot generate a *single* target image. In a different scenario, the user might want to try different target sizes and choose the one most suitable for his or her needs. Seam carving is linear in the number of pixels and resizing is therefore linear in the number of seams to be removed, or inserted. Nevertheless, when video is concerned carving one seam surface could take up to a few seconds. Therefore, computing seams in real time is a challenging task.

To address these issues we define a representation of multisize media that encodes, for an image or video of size $(m \times n)$, an entire range of retargeting sizes from 1×1 to $m \times n$ and even further for expansion to $N' \times M'$, when $N' > n$, $M' > m$ (see Section 4.3). Using a preprocessing stage to compute and encode the seams, multisize images and video allow real time retargeting to any target size even on very low end processing units.

Looking at it from a different perspective, multisize media can be seen as storing an *explicit* representation of the time-evolution implicit process of seam removals and insertions. Consider, for example, the case of changing the width of the media. We define an index map $\mathbf{V}$ of size $n \times m$ ($\times t$ if it is a video), that encodes, for each pixel, the index of the seam that removed it, i.e., $\mathbf{V}(i, j) = t$ means that pixel (i, j) was removed by the tth seam removal. To get an image (or frame in the video) of width m', we only need to gather, in each row, all pixels with seam index greater than or equal to $m - m'$. A similar map $\mathbf{H}$ can be defined to change the height of the media (Figure 9).

Figure 9: An image with its vertical and horizontal seam index maps V and H, colored by their index from blue (first seams) to red (last seams).



4. APPLICATIONS

In this section we demonstrate a number of applications based on seam carving.

4.1. Aspect ratio change

Assume we want to change the aspect ratio of a given image **I** from $n \times m$ to $n \times m'$ where $m - m' = c$. This can be achieved simply by successively removing c vertical seams from **I**. Contrary to simple scaling, this operation will not alter important parts of the image (as defined by the energy function), and in effect creates a nonuniform, content-aware resizing of the image (Figure 7).

The same aspect ratio correction, from $n \times m$ to $n \times m'$, can also be achieved by increasing the number of rows by a factor of m/m' (Figure 10). The added value of such an approach is that it does not remove any information from the image. We discuss our strategy for *increasing* an image size in details in Section 4.3.

Figure 10: Aspect ratio change of pictures of the Japanese master Utagawa Hiroshige. In both examples the original image is enlarged by seam insertion.

4.2. Image retargeting

In case the user is interested in changing both the width and the height of the image, this can be achieved by finding the optimal sequence of horizontal and vertical seams. User input can be provided in the form of positive pixel weights to guide the retargeting process. Figure 4 demonstrates automatic image retargeting, while Figure 11 shows a user guided example.

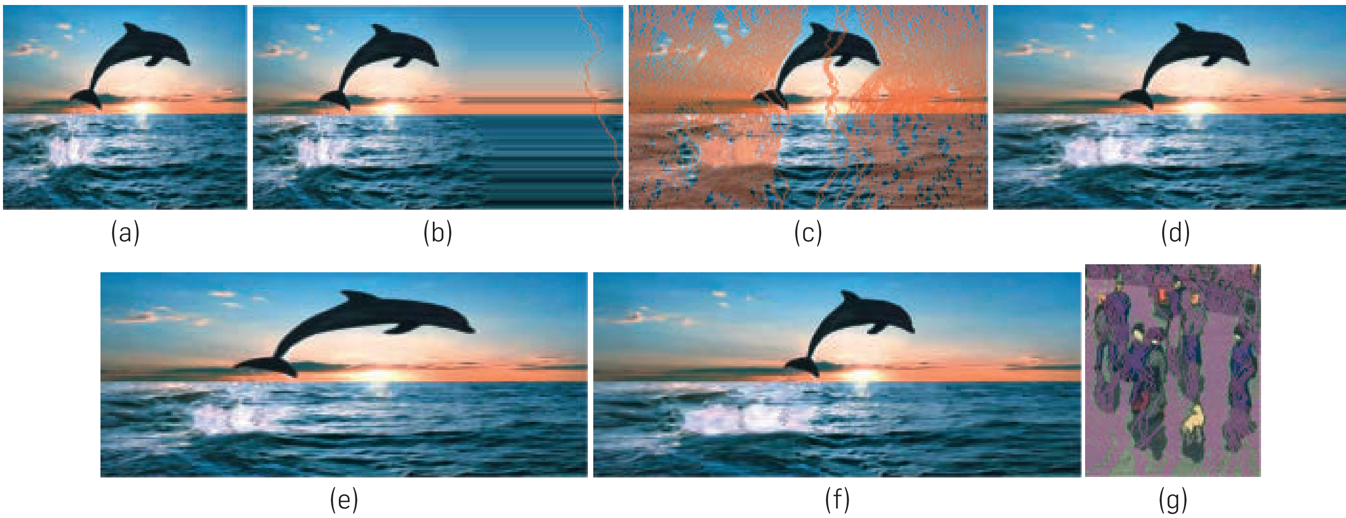
4.3. Image enlarging

The process of removing vertical and horizontal seams can be seen as a time-evolution process. We denote $\mathbf{I}^{(t)}$ as the smaller image created after t seam have been removed from **I**. To enlarge an image we approximate an "inversion" of this time evolution and insert new "artificial" seams to the image. Hence, to enlarge the size of an image **I** by one we compute the optimal vertical (horizontal) seam **s** on **I** and duplicate the pixels of **s** by averaging them with their left and right neighbors (top and bottom in the horizontal case).

Using the time-evolution notation, we denote the resulting image as $\mathbf{I}^{(-1)}$. Unfortunately, repeating this process will most likely create a stretching artifact by choosing the same seam (Figure 12(b)). To achieve effective enlarging, it is important to balance between the original image content and the artificially inserted parts. Therefore, to enlarge an image by k, we find the first k seams for *removal*, and duplicate them to arrive at $\mathbf{I}^{(-k)}$ (Figure 12(c)). This can be viewed as the process of traversing back in time to recover pixels from a larger image that would have been removed by seam removals (although it is *not* guaranteed to be the case).

Figure 11: Retargeting the Buddha. At the top is the original image, a cropped version where the ornaments are gone, and a scaled version where the content is elongated. Using simple bottom up feature detection for automatic retargeting cannot protect the structure of the face of the Buddha (bottom, left) and this is a challenging image for face detectors as well. By adding simple user constraints to protect the face (bottom, middle) or the face and flower (bottom, right), better results are achieved.

Figure 12: Seam insertion: finding and inserting the optimum seam on an enlarged image will most likely insert the same seam again and again as in (b). Inserting the seams in order of removal (c) achieves the desired 50% enlargement (d). Using two steps of seam insertions of 50% in (f) achieves better results than scaling (e). In (g), a close view of the seams inserted to expand Figure 10 is shown.



Duplicating all the seams in an image is equivalent to standard scaling (see Figure 12(e)). To continue in content-aware fashion for excessive image enlarging (for instance, greater than 50%), we break the process into several steps. Each step does not enlarge the size of the image in more than a fraction of its size from the previous step, essentially guarding the important content from being stretched. Nevertheless, extreme enlarging of an image would most probably produce noticeable artifacts (Figure 12(f)).

4.4. Object removal

We use a simple user interface for object removal. The user marks the object to be removed using negative weights, in effect drawing the seams to pass through these pixels. Consequently, seams are removed from the image until all marked pixels are gone. The system can automatically calculate the smaller of the vertical or horizontal diameters (in pixels) of the target removal region and perform vertical or horizontal removals accordingly (Figure 13). Moreover, to retain the original size of the image, seam insertion is employed on the resulting (smaller) image (see Figure 14). Note that this scheme alters the whole image (either its size or its content if it is resized back). This is because both the removed and inserted seams may pass anywhere in the image.

The reader is referred to more examples of retargeting and resizing of video and images at

http://www.faculty.idc.ac.il/Arik/SCWeb

Figure 13: Simple object removal: the user marks a region for removal (green), and possibly a region to protect (red), on the original image (see inset in left image). On the right image, consecutive vertical seam were removed until no "green" pixels were left.

5. LIMITATIONS

Most examples shown in this paper were computed automatically using the e_1 error function. However, it is clear that this scheme does not work well on all images. Other types of importance functions either manual or automatic could be used in combination with higher level cues such as face detectors to achieve better results (Figure 11).

Still, there are times when not even high level information can solve the problem. We can characterize two major factors that limit the seam carving approach. The first is the amount of content in an image. If the image is too condensed, in the sense that it does not contain "less important" areas, then any type of content-aware resizing strategy will not succeed. The second type of limitation is the layout of the image content. In certain types of images, albeit not being condensed, the content is laid out in a manner that prevents the seams from bypassing important parts (Figure 15).

Figure 14: Object removal: find the missing shoe (original image is top left). In this example, in addition to removing the object (one shoe), the image was enlarged back to its original size. Note that this example would be difficult to accomplish using in-painting or texture synthesis.

Figure 15: An example where seam carving fails because the image is too condensed. In such cases the best strategy might be to use scaling.

6. CONCLUSIONS AND FUTURE WORK

We presented an operator for content-aware resizing of images and video using seam carving. Seams are computed as the optimal paths on an image or video and are either removed or inserted. This operator can be used for a variety of image and video manipulations including aspect ratio change, image and video retargeting and object removal. The operator can be easily integrated with various saliency measures, as well as user input, to guide the resizing process. In addition, we define a novel media representation called multisize images and video that support continuous resizing ability in real time.

There are numerous possible extensions to this work. One would be to investigate better importance function that produces visually more pleasing results. Another direction is to rely on the seam map created by our method as a saliency map that can guide other processes such as content-aware compression. Finally, one could think of ways to unify the seam carving operator with other types such as scaling and cropping.

Acknowledgments

We thank Micheal Rubinstein for his work on video retargeting, Eric Chan for the use of the waterfall image and numerous flickr (*http://www.flickr.com/*) members for making their images available through creative common rights (*http://creativecommons.org/*): crazyegg95 (Buddha), Gusthy (Couple and Surfers), mykaul (Car), 02ma (Vase).

References

1. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M. Interactive digital photomontage. *ACM Trans. Graph. 23*, 3 (2004), 294–302.
2. Avidan, S. and Shamir, A. Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3 (2007), 10.
3. Boykov, Y. and Jolly, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *International Conference on Computer Vision (ICCV)*, volume I, 2001, 105–112.
4. Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. A visual attention model for adapting images on small displays. *Multimedia Systems 9*, 4 (2003), 353–364.
5. Davis, J. Mosaics of scenes with moving objects. In *Proceedings of CVPR*, 1998.
6. Efros, A.A. and Freeman, W.T. Image quilting for texture synthesis and transfer. In *SIGGRAPH 2001, Computer Graphics Proceedings*. E. Fiume, ed. ACM/ACM SIGGRAPH, 2001, 341–346.
7. Fan, X., Xie, X., Zhou, H.-Q., and Ma, W.-Y. Looking into video frames n small displays. In *MULTIMEDIA '03: Proceedings of the 11th ACM International Conference on Multimedia*. ACM, 2003, 247–250.
8. Gal, R., Sorkine, O., and Cohen-Or, D. Feature-aware texturing. In *Eurographics Symposium on Rendering*, 2006.
9. Itti, L., Koch, C., and Neibur, E. A model of saliency-based visual attention for rapid scene analysis. *PAMI 20*, 11 (1999), 1254–1259.
10. Jia, J., Sun, J., Tang, C.-K., and Shum, H.-Y. Drag-and-drop pasting. In *Proceedings of SIGGRAPH*, 2006.
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., and Bobick, A. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (2003), 277–286.
12. Liu, F. and Gleicher, M. Automatic image retargeting with fisheye-view warping. In *ACM UIST*, 2005, 153–162.
13. Liu, F. and Gleicher, M. Video retargeting: automating pan and scan. In *MULTIMEDIA'06: Proceedings of the 14th annual ACM International Conference on Multimedia*. ACM, 2006, 241–250.
14. Liu, H., Xie, X., Ma, W., and Zhang, H. Automatic browsing of large pictures on mobile devices. *Proceedings of the 11th ACM International Conference on Multimedia*, 2003, 148–155.
15. Roditty, L. and Zwick, U. On dynamic shortest paths problems. In *Proceedings of the 12th Annual European Symposium on Algorithms (ESA)*, 2004, 580–591.
16. Rother, C., Bordeaux, L., Hamadi, Y., and Blake, A. Autocollage. In *Proceedings of SIGGRAPH 2006*, 2006.
17. Rubinsteing, M., Shamir, A., and Avidan, S. Improved seam carving for video retargeting. *ACM Trans. Graph. 27*, 3 (2008), 10.
18. Santella, A., Agrawala, M., DeCarlo, D., Salesin, D., and Cohen, M. Gaze-based interaction for semi-automatic photo cropping. In *ACM Human Factors in Computing Systems (CHI)*, 2006, 771–780.
19. Schödl, A., Szeliski, R., Salesin, D. H., and Essa, I. Video textures. In *SIGGRAPH'00: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. ACM/Addison-Wesley, 2000, 489–498.
20. Setlur, V., Takagi, S., Raskar, R., Gleicher, M., and Gooch, B. Automatic image retargeting. In *The Mobile and Ubiquitous Multimedia (MUM)*. ACM, 2005.
21. Simakov, D., Caspi, Y., Shechtman, E., and Irani, M. Summarizing visual data using bidirectional similarity. In *Proceedings of CVPR*, 2008.
22. Suh, B., Ling, H., Bederson, B.B., and Jacobs, D.W. Automatic thumbnail cropping and its effectiveness. In *UIST'03: Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, 95–104.
23. Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
24. Wang, J., Bhat, P., Colburn, R. A., Agrawala, M., and Cohen, M. F. Interactive video cutout. *ACM Trans. Graph. 24*, 3 (2005), 585–594.
25. Wang, J., Reinders, M., Lagendijk, R., Lindenberg, J., and Kankanhalli, M. Video content presentation on tiny devices. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, 2004, 1711–1714.
26. Wang, J., Xu, Y., Shum, H.-Y., and Cohen, M. F. Video tooning. *ACM Trans. Graph. 23*, 3 (2004), 574–583.
27. Wang, Y.-S., Tai, C.-L., Sorkine, O., and Lee, T.-Y.. Optimized scale-and-stretch for image resizing. *ACM Trans. Graph. Proceedings of ACM SIGGRAPH ASIA 27*, 5 (2008).
28. Wei, L.-Y., Han, J., Zhou, K., Bao, H., Guo, B., Shum, H.-Y.. Inverse texture synthesis. *ACM Trans. Graph. 27*, 3 (2008), 1–9.
29. Wolf, L., Guttmann, M., and Cohen-Or, D. Non-homogeneous content-driven video-retargeting. In *Proceedings of the 11th IEEE International Conference on Computer Vision (ICCV'07)*, 2007, 1–6.

Ariel Shamir (arik@idc.ac.il), Efi Arazi School of Computer Science, The Interdisciplinary Center, Herzliya, Israel.

Shai Avidan (avidan@adobe.com), Adobe Systems, Inc., Newton, MA.

DOI:10.1145/1435417.1435438

Technical Perspective
Finding and Telling Stories with Data

By Jock D. Mackinlay

VISUAL ANALYSIS, A powerful method for finding and telling stories with data, is moving from research into widespread use. The research began in the 1960s with graphical user interfaces (GUIs), which supplanted command line interfaces by exploiting the power of the human visual/motor system. In the mid-1980s, advances in computer graphics hardware prompted research on visualization, the use of interactive, visual representations of data to amplify cognition. The early focus of visualization research was on individual analysts trying to find stories with data, first in the area of scientific data and then more generally with abstract information.[1]

Since 2000, the research focus has expanded from the visualization of an individual analyst to visual analysis—the use of visualization in larger processes of sensemaking.[2] Reduced to its essence, visual analysis has a four-part cycle:

- Focus on a data-oriented task,
- Forage for relevant data,
- Visualize the data, and
- Perform an appropriate action.

Given a task, analysts forage for relevant data, which is mapped to visualizations that exploit the power of the human visual system. Visualizations lead to findings, which prompt actions. When the actions are new data analysis tasks, the cycle repeats. There are also internal cycles in this problem-solving process. For example, visualizations can indicate the need to forage for new data.

Voyagers and Voyeurs: Supporting Asynchronous Collaborative Visualization by Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg further expanded the research focus into the area of collaborative visualization, in which telling stories with data plays a central role. In Part 4 of the visual analysis cycle, most analysts must collaborate with colleagues and managers before actions are approved. Data views make data understandable, which encourages collaboration with people who are not skilled analysts. Interactive data views allow people to do their own analysis with data views authored by others.

The authors describe a prototype Web application that includes several techniques for supporting collaborative visualization, and report on user studies involving the prototype.

The most interesting aspect of the prototype is a bookmarking mechanism that supports doubly linked discussions. Data views have the property that the same view can be specified in multiple ways. The authors describe how to associate a bookmark with a data view rather than the various specifications of the data view, which supports asynchronous discussions about views.

The most interesting aspect of the user studies was that their subjects switched between data-driven exploration and social navigation—that is, between being data voyagers and data voyeurs.

This paper represents an important early step in research on collaborative visualization. The authors made the excellent choice to focus their prototype on the U.S. census data set. By focusing on a single public domain data set, they reduced their prototype's data foraging complexity, thus encouraging their users to focus on collaborative activities. Furthermore, many people are interested in census data.

The next step for research on collaborative visualization is to address topics that arise in more fully featured visual analysis applications. For example, most visual analysis tasks involve multiple data sources. Unlike the U.S. census data set, many data sources have security issues that make collaboration more difficult. Some data sources also change rapidly, making asynchronous conversations more complex. Finally, tasks involving multiple data sources often require conversations that compare and contrast multiple data views that must be viewed simultaneously.

Finding and telling stories with data can help people understand the world more clearly. For example, the mortgage-backed security crisis might have been averted if mortgage data had been available for storytelling. The key is to have visual analysis technology and use it appropriately.

References

1. Card, S.K., Mackinlay, J.D., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think.* Morgan Kauffman, 1999.
2. Russell, D.M., Stefik, M.J., Pirolli, P., and Card, S.K. The cost structure of sensemaking. In *Proceedings of the ACM Conference of Human-Computer Interactions* (1993), 269–276.

Jock D. Mackinlay (jmackinlay@tableausoftware.com) is the director of Visual Analysis at Tableau Software, Seattle, WA.

DOI:10.1145/1435417.1435439

Voyagers and Voyeurs: Supporting Asynchronous Collaborative Visualization

By Jeffrey Heer, Fernanda B. Viégas, and Martin Wattenberg

Abstract

This article describes mechanisms for asynchronous collaboration in the context of information visualization, recasting visualizations as not just analytic tools, but social spaces. We contribute the design and implementation of *sense.us*, a Web site supporting asynchronous collaboration across a variety of visualization types. The site supports view sharing, discussion, graphical annotation, and social navigation and includes novel interaction elements. We report the results of user studies of the system, observing emergent patterns of social data analysis, including cycles of observation and hypothesis, and the complementary roles of social navigation and data-driven exploration.

1. INTRODUCTION

Visual representations of information often lead to new insights by enabling viewers to see data in context, observe patterns, and make comparisons. In this way, visualizations leverage the human visual system to improve our ability to process large amounts of data. Card et al.[6] describe how visualization supports the process of *sensemaking*, in which information is collected, organized, and analyzed to form new knowledge and inform further action. They emphasize the ways visualization exploits an individual's visual perception to facilitate cognition.

In practice, however, sensemaking is often also a social process. People may disagree on how to interpret the data and may contribute contextual knowledge that deepens understanding. As participants build consensus or make decisions they learn from their peers. Furthermore, some data sets are so large that thorough exploration by a single person is unlikely. This suggests that to fully support sensemaking, visualizations should also support social interaction. In this spirit, a recent report[23] names the design of collaborative visualization tools as a grand challenge for visualization research.

These considerations are not just hypothetical. For example, the manager of a business group in our company described to us how quarterly reports are disseminated within his organization via e-mail. Heated discussion takes place around charts and graphs as the group debates the causes of sales trends and considers possible future actions. However, writing about particular trends or views is difficult, involving awkward references to attached spreadsheets from the e-mail text. Furthermore, the discussion is scattered and disconnected from the visualizations, making it difficult for newcomers to catch up or others to review and summarize the discussion thus far. According to the manager of the group, the analysis process could benefit from a system for sharing, annotating, and discussing the visualized data.

Similar scenarios appear in other domains. Moreover, experiences with deployments of visualizations hint at ways that social phenomena already occur around visualizations. Wattenberg and Kriss[27] describe the response to NameVoyager, an online visualization of historical baby name trends. Playful yet often surprisingly deep analysis appeared on numerous blogs as participants discussed their insights and hypotheses. Observing the use of a physical installation of the Vizster social network visualization, Heer[18] noted that groups of users, spurred by storytelling of shared memories, spent more time exploring and asked deeper analysis questions than individuals. Similarly, Viégas et al.[24] found that users of the PostHistory e-mail archive visualization immediately wanted to share views with friends and family and engage in storytelling.

While suggestive, these observations provide only a circumstantial understanding of the social aspects of asynchronous analysis around visualizations. In the case of the NameVoyager and PostHistory, the findings were essentially accidental. Vizster was designed for playful interaction, but in a synchronous and less analytic context. It would therefore be valuable to replicate these findings to deepen our understanding of this type of interaction.

Furthermore, if social interaction is an important accompaniment to data visualization, it is natural to look for ways to support and encourage it. To address both these goals, we designed and implemented a Web site, *sense.us*, aimed at group exploration of demographic data. The site provides a suite of interactive visualizations and facilitates collaboration through view bookmarking, doubly linked discussions, graphical annotation, saved bookmark trails, and social navigation through comment listings and user profiles. We then conducted user studies to observe closely how people engage in social data analysis. The studies also allowed us to evaluate the new design elements in the site and suggest directions for future work.

A previous version of this paper was published in the *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, April 2007.

2. PRIOR WORK

Collaboration has been well studied in contexts that are not directly related to information visualization. The study of how computer systems can enable collaboration is referred to as *computer-supported cooperative work*, or CSCW. Because collaboration occurs in a variety of situations, CSCW scholars often use a "time-space" matrix[21] to outline the conceptual landscape. The time dimension represents whether or not participants interact at the same time (synchronously or asynchronously)—for example, instant messaging is a largely synchronous communication medium, while e-mail is asynchronous. The space dimension describes whether users are collocated or geographically distributed.

Most work on collaborative visualization has been done in the context of synchronous scenarios: users interacting at the same time to analyze scientific results or discuss the state of a battlefield. Collocated collaboration usually involves shared displays, including wall-sized, table-top, or virtual reality displays (e.g., Dietz,[14] General Dynamics[16]). Systems supporting remote collaboration have primarily focused on synchronous interaction,[1,4] such as shared virtual workspaces[8] and augmented reality systems that enable multiple users to interact concurrently with visualized data.[3,9] In addition, the availability of public displays has prompted researchers to experiment with asynchronous, collocated visualization (same place, different time), for example, in the form of ambient displays that share activity information about collocated users.[7]

In this article, we focus on remote asynchronous collaboration—the kind of collaboration that is most common over the Web. One reason for our interest is that partitioning work across both time and space holds the potential of greater scalability in group-oriented analysis. For example, one decision-making study found that asynchronous collaboration resulted in higher-quality outcomes—broader discussions, more complete reports, and longer solutions—than face-to-face collaboration.[2] However, as noted by Viégas and Wattenberg,[25] little research attention has been dedicated to asynchronous collaboration around interactive visualization. Instead, users often rely on static imagery when communicating about these interactive systems. Images of the visualization are transferred as printouts or screenshots, or included in word-processing or presentation documents.

A few commercial visualization systems introduced prior to our work provide asynchronous collaboration features. Online mapping systems (e.g., Google Maps) provide bookmarks (URLs) that users can send to others to share views. The visualization company Spotfire provides DecisionSite Posters, a Web-based system that allows a user to post an interactive visualization view that other users can explore and comment on. The Posters apply only to a subset of Spotfire's full functionality and do not allow graphical annotations, limiting their adoption.[25]

One common feature of these systems is *application bookmarks*: URLs or URL-like objects that point back into a particular state of the application, for example, a location and zoom level in the case of Google Maps. This pattern is not surprising; for users to collaborate, they must be able to share what they are seeing to establish a common ground for conversation.[12]

One of the primary uses of bookmarks is in discussion forums surrounding a visualization. Some systems use what we term *independent discussion*, where conversations are decoupled from the visualization. For example, Google Earth provides threaded discussion forums with messages that include bookmarks into the visualized globe. In such systems there are unidirectional links from the discussion to the visualization, but no way to discover related comments while navigating the visualization itself.

Another stream of related work comes from wholly or partly visual annotation systems, such as the regional annotations in sites such as Flickr.com and Wikimapia.org and in Churchill et al.'s anchored conversations.[10] Such systems enable *embedded discussion* that places conversational markers directly within a visualization or document. Discussion of a specific item may be accessed through a linked annotation shown within the visualization. These systems may be seen as the converse of independent discussions, allowing unidirectional links from an artifact to commentary.

In this article, we extend the past work with a comprehensive design for asynchronous collaboration around interactive data visualizations, addressing issues of view sharing, discussion, graphical annotation, and social navigation.

3. THE DESIGN OF SENSE.US

To explore the possibilities for asynchronous collaborative visualization, we designed and implemented sense.us, a prototype Web application for social visual data analysis. The site provides a suite of visualizations of United States census data over the last 150 years (see Figures 1 and 2) and was designed for use by a general audience. We built sense. us to put our design hypotheses into a concrete form which we could then deploy and use to study collaborative data exploration.

The primary interface for sense.us is shown in Figure 1. In the left panel is a Java applet containing a visualization. The right panel provides a discussion area, displaying commentary associated with the current visualization view, and a graphical bookmark trail, providing access to views bookmarked by the user. With a straightforward bookmarking mechanism, sense.us supports collaboration with features described in detail below: doubly linked discussions, graphical annotations, saved bookmark trails, and social navigation via comment listings and user activity profiles.

3.1. View sharing

When collaborating around visualizations, participants must be able to see the same visual environment in order to ground[12] each others' actions and comments. To this aim, the sense.us site provides a mechanism for bookmarking views. The system makes application bookmarking transparent by tying it to conventional Web bookmarking. The browser's location bar always displays a URL that links to the current state of the visualization, defined by the settings of filtering, navigation, and visual encoding

Figure 1: The sense.us collaborative visualization system. (a) An interactive visualization applet, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to add a link to that view in the comment. (e) Threaded comments attached to the current view. (f) URL for the current state of the application. The URL is updated automatically as the visualization state changes.

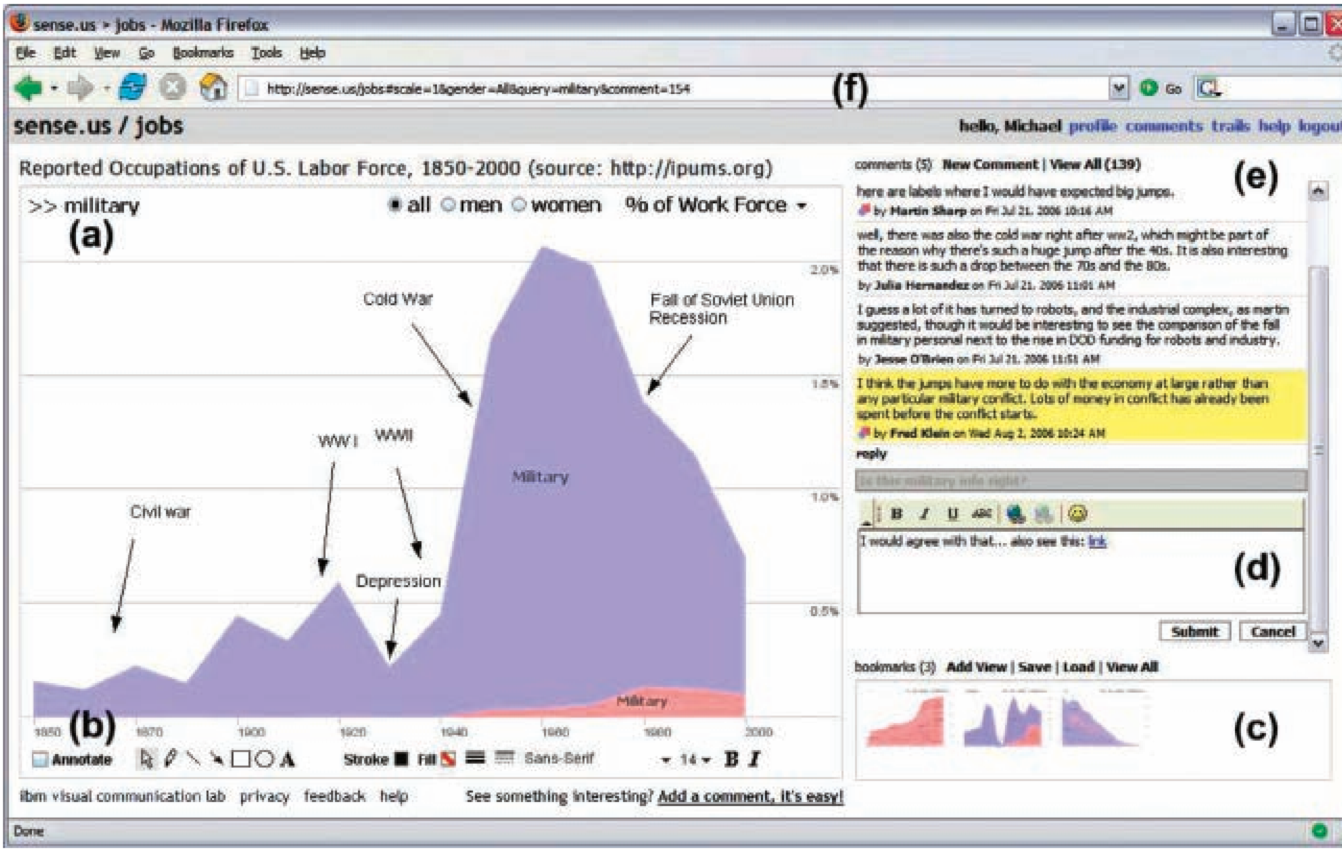
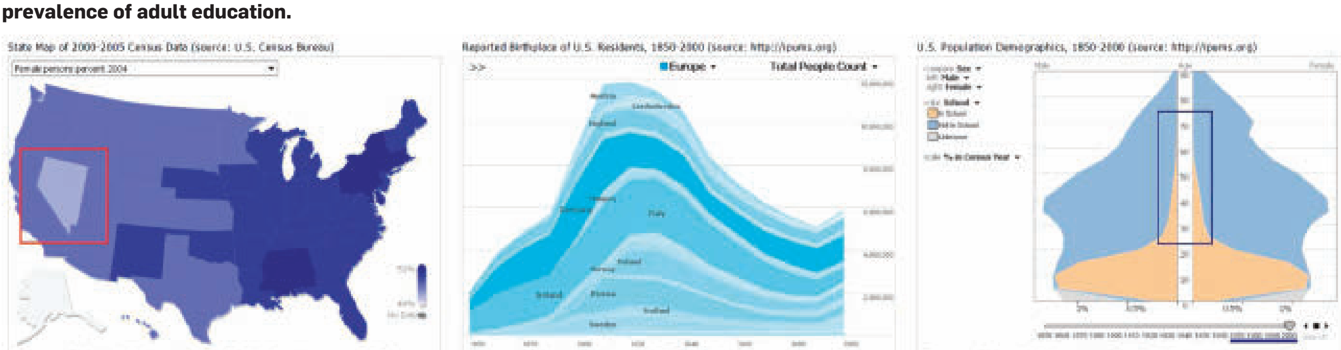


Figure 2: Sample visualizations from sense.us. (a) Interactive state map. The image shows the male/female ratio of the states in 2005. (b) Stacked time series of immigration data, showing the birthplace of U.S. residents over the last 150 years. The image shows the number of U.S. residents born in European countries. (c) Population pyramid, showing population variation across gender and age groups. Additional variables are encoded using stacked, colored bands. The image visualizes school attendance in 2000; an annotation highlights the prevalence of adult education.



parameters. As the visualization view changes, the URL updates to reflect the current state (Figure 1f), simplifying the process of sharing a view through e-mail, blogs, or instant messaging by enabling users to cut-and-paste a link to the current view at any time. To conform to user expectations, the browser's back and forward buttons are tied to the visualization state, allowing easy navigation to previously seen views.

3.2. Doubly linked discussion

To situate conversation around the visualization, we created a technique called *doubly linked discussion*. The method begins with an "independent" discussion interface in which users can attach comments to particular states (or views) of a visualization. Comments are shown on the right side of the Web page and grouped into linear discussion threads (Figure 1e). Each comment shows the thread topic, comment text, the author's full name, and the time at which the comment was authored. Clicking on a comment takes the visualization to a bookmarked state representing the view seen by the comment's author.

Users can add comments either by starting a new thread or posting a reply to an existing thread. When a "New Comment" or "Reply" link is clicked, a text editor appears at the site where the comment will be inserted and the graphical annotation tools (discussed next) become active. Upon submission, the comment text and any annotations are sent to the server and the comment listing is updated.

The interface described above is based on links from the commentary into the visualization. Our system also provides links in the other direction: from the visualization into the discussion. As a user changes parameters and views in the visualization, they may serendipitously happen upon a view that another person has already commented on. When this occurs, the relevant comments will automatically appear in the right-hand pane. Our intuition was that this "doubly linked" discussion interface, which combines aspects of independent and embedded discussion, would facilitate grounding and enable the visualization itself to become a social place.

3.3. Pointing via graphical annotation

In real-time collocated collaboration, participants commonly use both speech and gesture, particularly pointing,[11,20] to refer to objects and direct conversation. For asynchronous collaboration, graphical annotations can play a similar communicative role. We hypothesized that graphical annotations would be important both for pointing behavior and playful commentary. To add a pictorial element to a comment or point to a feature of interest, authors can use drawing tools (Figure 1b) to annotate the commented view. These tools allow free-form ink, lines, arrows, shapes, and text to be drawn over the visualization view. The tools are similar to presentation tools such as Microsoft PowerPoint and are intended to leverage users' familiarity with such systems.

Comments with annotations are indicated by the presence of a small shape logo to the left of the author's name in the comment listing (see Figure 1e). When the mouse hovers over an annotated comment, the comment region highlights in yellow and a hand cursor appears. Subsequently clicking the region causes the annotation to be shown and the highlighting to darken and become permanent. Clicking the comment again (or clicking a different comment) will remove the current annotation and highlighting.

We refer to this approach as *geometric annotation*, which operates like an "acetate layer" over the visualization, in contrast to *data-aware* annotations directly associated with the underlying data. We chose to implement a free-form annotation mechanism so that we could first study pointing behaviors in an unconstrained medium. Aside from the freedom of expression it affords, geometric annotation also has a technical advantage: it allows reuse of the identical annotation system across visualizations, easing implementation and preserving a consistent user experience.

3.4. Collecting and linking views

In data analysis it is common to make comparisons between different ways of looking at data. Furthermore, storytelling has been suggested to play an important role in social usage of visualizations, as discussed by Viégas et al.[24] Drawing comparisons and telling stories both require the ability to embed multiple view bookmarks into a single comment.

To support such multiview comments and narratives, we created a "bookmark trail" widget. The bookmark trail functions something like a shopping cart: as a user navigates through the site, he or she can click a special "Add View" link to add the current view to a graphical list of bookmarks (Figure 1c). Bookmarks from any number of visualizations can be added to a trail. A trail may be named and saved, making it accessible to others.

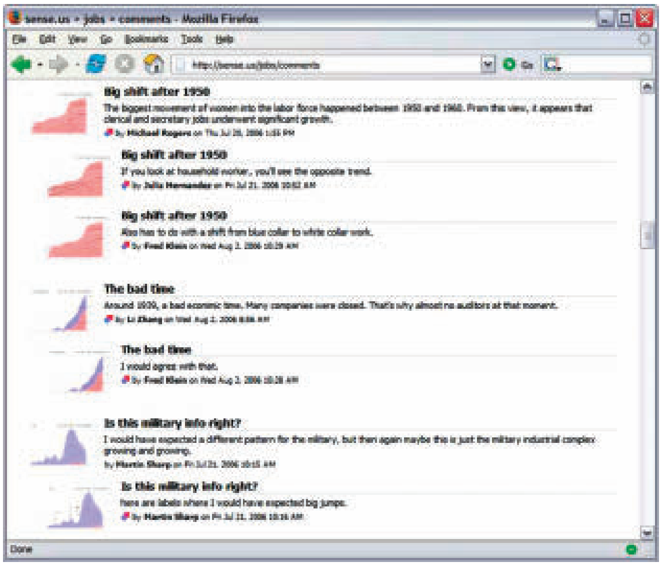
The bookmark trail widget also functions as a short-term storage mechanism when making a comment that includes links to multiple views. Dragging a thumbnail from the bookmark trail and dropping it onto the text area create a hyperlink to the bookmarked view; users can then directly edit or delete the link text within the text editor. When the mouse hovers over the link text, a tooltip thumbnail of the linked view is shown.

3.5. Awareness and social navigation

Social navigation[15] leverages usage history to provide additional navigation options within an information space. Our initial system supports social navigation through comment listings and user profile pages that display recent activity. Comment listings provide a searchable and sortable collection of all comments made within the system, and can be filtered to focus on a single visualization (see Figure 3). Comment listing pages include the text and a thumbnail image of the visualization state for each comment. Hovering over the thumbnail yields a tooltip with a larger image. Clicking a comment link takes the user to the state of the visualization where the comment was made, displaying any annotations included with the comment. The author's name links to the author's profile page, which includes their five most recent comment threads and five most recently saved bookmark trails. The view also notes the number of comments made on a thread since the user's last comment, allowing users to monitor the activity of discussions to which they contribute.

Although more elaborate social navigation mechanisms are possible, we wanted to observe system usage with just these basic options. We were particularly interested in observing the potential interplay between data-driven exploration and social navigation. By allowing discussions to be retrieved unobtrusively while a user explores the data,

Figure 3: The sense.us comment listing page. Comment listings display all commentary on visualizations and provide links to the commented visualization views.



potentially relevant conversation can be introduced into the exploration process. Meanwhile, comment listings and indications of recent posts may help users find views of interest, making social activity a catalyst for data exploration.

3.6. Unobtrusive collaboration

Finally, while designing sense.us we also wished to follow a common CSCW design guideline: collaborative features should not impede individual usage.[17] Hence we did not litter views with prior annotations or commentary. Rather, commentary on a visualization is retrieved and displayed unobtrusively on the right side of the screen and graphical annotations are displayed "on demand" by the user.

4. IMPLEMENTATION NOTES

While many aspects of sense.us rely on well-known techniques, this section provides implementation details for the more complex features: application bookmarking, doubly linked discussions, and graphical annotations.

4.1. Application bookmarking

Bookmarks of visualization state are implemented as a set of name–value pairs of visualization parameters, listed using standard URL query syntax. Normally, changing the browser's URL will force a reload of the page to prevent security attacks. Because a reload would cause a disruptive restart of the visualization applet, the bookmark URL encodes the query string as a page anchor—using the URL '#' delimiter instead of the standard '?' delimiter—so that the URL updates in place. Furthermore, updated URLs are put into the browser's history stack, so that the browser's back and forward buttons have their usual behavior. When a visualization URL is updated due to use of the back or forward buttons or manual typing, scripts send the updated URL to the applet, which is parsed and used to update the current visualization state.

4.2. Doubly linked discussions

The bookmarking mechanisms alone are not sufficient to support doubly linked discussions. To see the challenge in linking from a view state back to all comments on that view, consider the visualization in Figure 1. When a user types "military" into the top search box, they see all jobs whose titles begin with the string "military." On the other hand, if they type only "mili," they see all titles beginning with "mili"—but this turns out to be the identical set of jobs. These different parameter settings result in different URLs, and yet provide exactly the same visualization view. More generally, parameter settings may not have a one-to-one mapping to visualization states. To attach discussions to views we therefore need an indexing mechanism which identifies visualization states that are equivalent despite having different parametric representations.

We solve this indexing problem by distinguishing between two types of parameters: *filter parameters* and *view parameters*. Filter parameters determine which data elements are visible in the display. Rather than index filter parameters directly, we instead index the filtered state of the application by noting which items are currently visible, thereby capturing the case when different filter parameters give rise to the same filtered state. View parameters, on the other hand, adjust visual mappings, such as selecting a normalized or absolute axis scale. Our current system indexes the view parameters directly. The bookmarking mechanism implements this two-part index by computing a probabilistically unique hash value based on both the filtered state and view parameters. These hash values are used as keys for retrieving the comments for the current visualization state.

4.3. Annotation

The graphical annotations take the form of vector graphics drawn above the visualization. When a new comment is submitted, the browser requests the current annotation (if any) from the visualization applet. The annotation is saved to an XML format, which is then compressed using gzip and encoded in a base 64 string representation before being passed to the browser. When comments are later retrieved from the server, the encoded annotations are stored in the browser as JavaScript variables. When the user requests that an annotation be displayed, the encoded annotations are passed to the applet, decoded, and drawn.

5. EVALUATION

To gain a preliminary understanding of asynchronous collaboration practices around visualizations, we ran exploratory user studies of the sense.us system. The studies had two specific goals: first, to better understand emergent usage patterns in social data analysis; second, to learn how well the various features of the sense.us system supported this analysis. We ran the studies in two different parts: a pair of controlled lab studies and a 3-week live deployment on the IBM corporate intranet. To analyze the data, we employed a mixed-methods analysis approach combining qualitative and quantitative observations.

5.1. Lab study

We first ran a pilot study with 6 subjects (2 females, 4 males), all of whom were members of our immediate research team. Comments from the pilot were visible in a subsequent 12 subject (3 females, 9 males) study, with subjects drawn from our greater research lab. Subjects were at least peripherally familiar with each other and many were coworkers. Ages ranged from the early-twenties to mid-fifties and education varied from the undergraduate to the doctoral level, spanning backgrounds in computer science, design, social science, and psychology. Concerned that our lab's focus in collaborative software might bias results, we replicated the lab study in a university environment with additional 12 subjects (5 females, 7 males). Subject variation in age, education, and social familiarity remained similar.

Subjects conducted a 25 min usage session of the sense.us system. A single visualization was available in the study: a stacked time series of the U.S. labor force over time, divided by gender (Figure 1). Users could navigate the visualization by typing in text queries (matched to job title prefixes), filtering by gender, and setting the axis scale, either to total people count or percentage values.

This data set was chosen for several reasons. First, job choice is a topic that most of our users should have no difficulty relating to. Second, like many other real-world data sets, there are data collection issues, including missing data and unclear or antiquated labels. Third, we suspected the data would be an interesting boundary case for annotations, as for many visualization views, text seemed sufficient when referencing spikes or valleys in the data.

After a brief tutorial of system features, participants were instructed to use the system however they liked—no specific tasks were given. However, users were told that if they felt at a loss for action, they could browse the data for trends they found interesting and share their findings. An observer was present taking notes and a think-aloud protocol was used. User actions were also logged by the software. Subjects were run in sequential order, such that later participants could view the contributions of previous subjects but not vice versa. The system was seeded with five comments, each with an observation of a particular data trend.

After the study, subjects completed a short exit questionnaire about their experiences. Participants were asked to rate on a 5-point Likert scale to what degree (1) they enjoyed using the system, (2) they learned something interesting, (3) others' comments were helpful in exploring the data, and if they found annotations useful for (4) making their own comments, or (5) understanding others' comments. Subjects were also asked free response questions about what they liked, disliked, and would change about the system.

5.2. Live deployment

We also conducted a live deployment of the system on the IBM corporate intranet for 3 weeks. Any employee could log in to the system using their existing intranet account. Eight visualizations were available in the system, among them were the visualizations of Figures 1 and 2 and a scatterplot of demographic metrics (see Figure 4). We also introduced two visualizations specific to the company: stacked time series of keyword tagging activity and individual user activity on *dogear*, an internal social bookmarking service. The site was publicized through an e-mail newsletter, an intranet article, and individual e-mails.

5.3. Findings

In the rest of this section, we report observations from these studies, organized by commentary, graphical annotations, navigation patterns, and use of doubly linked discussion. As variation in content and tone differed little across studies, the discussion incorporates data aggregated from each. The data analyzed were drawn from 12.5 h of qualitative observation and from usage logs including 258 comments: 41 from the pilot, 85 from the first study, 60 from the second, and 72 from the deployment.

5.4. Comments

We first wanted to learn how comments were being used to conduct social data analysis—was there a recognizable structure to the discussions? To find out, we performed a formal content analysis on the collected comments. Each paper author independently devised a coding rubric based upon a reading of the comments. We then compared our separate rubrics to synthesize a final rubric that each author used to independently code the comments. The final coding rubric categorized comments as including zero or more of the following: observations, questions, hypotheses, links or references to other views, usage tips, socializing or joking, affirmations of other comments, to-dos for future actions, and tests of system functionality. We also coded whether or not comments made reference to data naming or collection issues, or to concerns about the Web site or visualization design. The coded results were compared using Cohen's

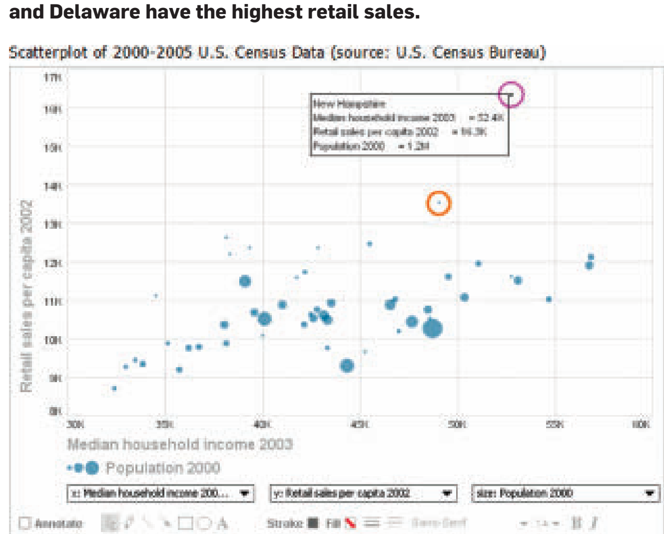


Figure 4: Scatterplot of U.S. states showing median household income (*x*-axis) vs. retail sales per capita (*y*-axis). New Hampshire and Delaware have the highest retail sales.

kappa statistic. The lowest pairwise kappa value was 0.74, indicating a satisfactory inter-rater reliability.

Most commentary on sense.us involved data analysis. A typical comment made note of an observed trend or outlier, often coupled with questions, explanatory hypotheses, or both. A typical reply involved discussing hypotheses or answering questions. The results of coding the comments are shown in Figure 5. In total, 80.6% of comments involved an observation of visualized data, 35.5% provided an explanatory hypothesis, and 38.1% included a question about the data or a hypothesis. Most questions and hypotheses accompanied an observation (91.6% and 92.2%, respectively) and half the hypotheses were either phrased as or accompanied by a question (49.0%).

For example, participants in both lab studies discovered a large drop in bartenders around the 1930s and posted comments attributing the drop to alcohol prohibition. In the live deployment, one user commented on a scatterplot view, asking why New Hampshire has such a high level of retail sales per capita (Figure 4). Another user noted that New Hampshire does not have sales tax, and neither does Delaware, the second highest in retail sales. In this fashion, discussion regularly involved the introduction of contextual information not present in the visualization. For instance, Figure 1 includes a timeline of events that was iteratively constructed by multiple users, while the graph of teachers in Figure 6 notes the introduction of compulsory education.

One instance of social data analysis occurred around a rise, fall, and slight resurgence in the percentage of dentists in the labor force. The first comment (one of the five seed comments) noted the trends and asked what was happening. One subject responded in a separate thread, *"Maybe this has to do with fluoridation? But there's a bump . . . but kids got spoiled and had a lot of candy??"* To this another subject responded *"As preventative dentistry has become more effective, dentists have continued to look for ways to continue working (e.g., most people see the dentist twice a year now v. once a year just a few decades ago)."* Perhaps the most telling comment, however, included a link to a different view, showing both dentists and dental technicians. As dentists had declined in percentage, technicians had grown substantially, indicating specialization within the field. To this, another user asked *"I wonder if school has become too expensive for people to think about dentistry, or at least their own practice when they can go to technical school for less?"* Visual data analysis, historical knowledge, and personal anecdote all played a role in the sensemaking process, explicating various factors shaping the data.

Another role of comments was to aid data interpretation, especially in cases of unclear meaning or anomalies in data collection. Overall, 15.7% of comments referenced data naming, categorization, or collection issues. One prominent occupation was labeled "Operative," a general category consisting largely of skilled labor. This term had little meaning to subjects, one of whom asked *"what the hell is an operative?"* Others responded to reinforce the question or to suggest an explanation, *e.g., "I bet they mean factory worker."* Another subject agreed, noting that the years of the rise and fall of operatives seemed consistent with factory workers. Other examples include views missing data for a single year (1940 was a common culprit), leading users to comment on the probable case of missing data.

Figure 5: Content analysis categorization of sense.us comments. Categories are not mutually exclusive.

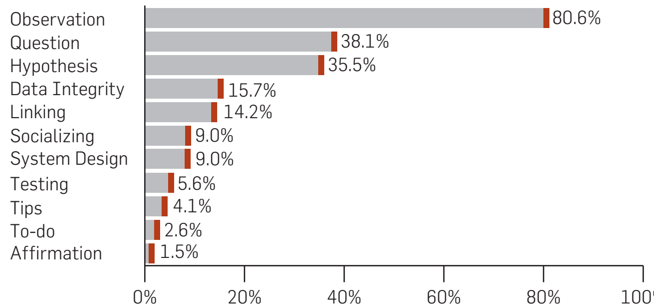
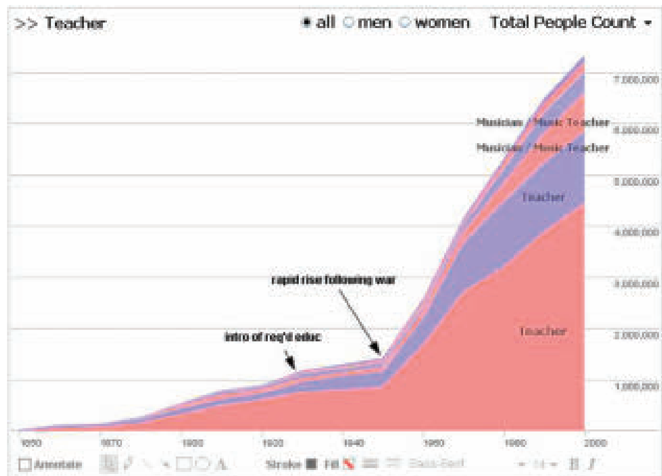


Figure 6: Visualization of the number of teachers. Annotations indicate the start of compulsory education and the rise of teachers in the post-World War II era.



Some users were less interested in specific views than in recurring patterns. One user was interested in exploring careers that were historically male-dominated, but have seen increasing numbers of females in the last half-century. The user systematically explored the data, saving views in a bookmark trail later shared in a comment named *"Women's Rise."* Similarly, a more mathematically minded participant was interested in patterns of job fluctuations, creating a trail showcasing recurring distributions. Another searched for jobs that had been usurped by technology, such as bank tellers and telephone operators. In each of these cases, the result was a tour or story winding through multiple views.

Overall, 14.2% of comments referenced an additional view, either implicitly in the text or explicitly through drag-and-drop bookmark links. Although 22 of the 24 lab study subjects (87.5%) saved at least one view to the bookmark trail, only 14 (58.3%) created one or more drag-and-drop bookmark links. The amount of view linking varied by user, ranging from 0 to 19 links with an average of 2.17.

Comments served other purposes as well. A number were simple tests of system functionality (5.6%), often deleted by the user. Some included tips for using the system (4.1%), noting how to take advantage of specific features. Overall, 9.0% of comments referenced the site design, either in the form of usage tips or feature requests. A few comments included to-dos for future work (2.6%), such as later adding a link to a relevant wikipedia article. Others served solely as affirmations to another comment (1.5%). For example, people stating *"I agree with that"* to support a hypothesis. In many cases, study participants would note out loud *"that is interesting!"* without posting a comment to the system.

Finally, some comments were social in nature (9.0%). Most pointed out trends in the data, but did so in a joking manner. One user built a view comparing female lawyers and bartenders, writing *"Women at the bar and behind the bar."* In the pilot study, one of our lab members annotated a drop in stock brokers after 1930 with a picture of a person's trajectory off a skyscraper (Figure 7). This elicited smiles and laughter from subjects in the subsequent study, one of whom replied with an affirmation simply saying *"Whoa!"*

We also analyzed the structural aspect of comments. Excluding comments from the pilot study, deleted test comments, and those written by the paper authors, 195 comments were collected. Of those, 140 (71.8%) started new discussion threads while 55 (28.2%) were replies to existing threads. The average thread length was 1.35 comments (*SD* 0.82), with a maximum of 5 comments. In some cases, discussion spanned multiple threads.

5.5. Graphical annotation

Next, we wanted to understand how graphical annotations were used and to what degree they contributed to social data analysis. Of the 195 nonpilot, nondeleted comments, 68 (35.9%) included annotations. The vast majority (88.6%) of annotations involved pointing to items or trends of interest. The others (11.4%) involved more playful expression, such as drawn smiley faces and the visual commentary of Figure 7.

Across these annotations, a total of 179 "shapes" were drawn, with the options being free-form ink, lines, arrows, rectangles, ovals, and text. Arrows were the most popular shape (25.1% of shapes), and were used to point to items as well as to situate information provided by text captions (24.6%). Ovals (17.9%) were primarily used to enclose regions of interest. Free-form ink drawn with the pencil tool (16.2%) was used for pointing, enclosing irregularly shaped regions, and free-form drawing. Of the rest, lines made up 14.5% of all shapes and rectangles only 1.7% (Figure 8).

A few users, particularly those with experience in graphic design, noted that graphical annotations were their favorite feature. Other users noted that the annotations were often unnecessary for comments where text could describe the trend(s) of interest. A few of these users added annotations to such views anyway, saying the annotations were *"surprisingly satisfying,"* enabling *"personal expression."* Exit survey results somewhat reflected these views, as users ranked annotations more useful for writing their own comments ($M = 3.5/5.0$, $SD = 0.85$) than understanding others' comments ($M = 3.2/5.0$, $SD = 0.90$). This difference, however, did not reach statistical significance ($t(23) = -1.67$, $p < 0.108$, two-tailed).

5.6. Visitation and navigation

Our next questions concerned how users navigated the visualizations. Most users began exploring the data directly, starting from the default overview and drilling down. A few immediately went to the comments listing to see what others had done. Many participants searched for their own occupations and those of friends and family. Other strategies included browsing for items of interest found in the overview (*"Wow, look how the poor farmers died out"*) and formulating queries based on an over-arching interest, such as gender balance.

Looking to the usage logs, navigation by interaction with the visualization or attached commentary was by far the most common navigation technique, accounting for 70.5% of state views. The second most popular was the back and forward buttons at 17.5%, validating our integration of the visualization with browser history mechanisms. Following a link from the comment listings accounted for 8.7% of all views, while the final 3.3% were due to clicking a bookmark in the bookmark trail (Figure 9).

Figure 7: Annotated view of stock brokers. The attached comment reads "Great depression 'killed' a lot of brokers."

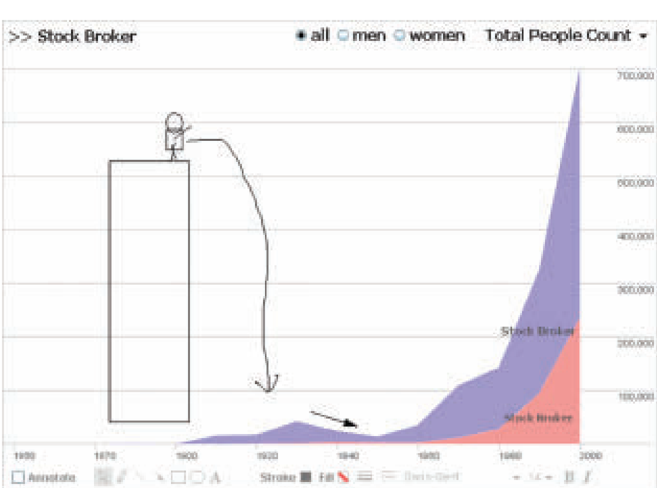


Figure 8: Usage of sense.us graphical annotation tools.

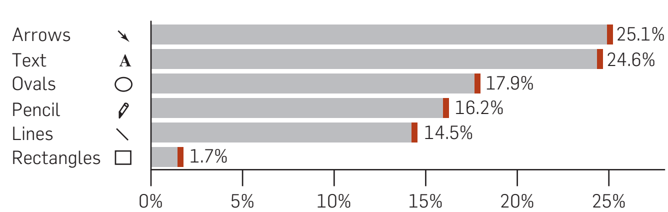
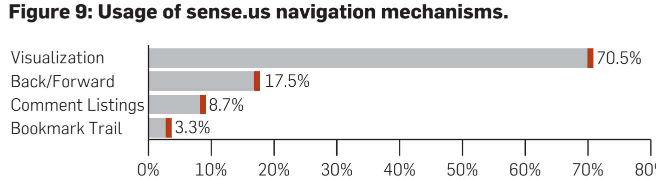


Figure 9: Usage of sense.us navigation mechanisms.



At some point, every subject explored the comment listings. Some felt they would find interesting views more quickly. Remarks to this effect included *"I bet others have found even more interesting things"* and *"You get to stand on the shoulders of others."* Other subjects were interested in specific people they knew or discovering what other people had investigated. Said one participant, *"I feel like a data voyeur. I really like seeing what other people were searching for."* Switching between data-driven exploration and social navigation was common. Views discovered via comment listings often sparked new interests and catalyzed more data-driven exploration. After some exploration, participants routinely returned to the listings for more inspiration. In the survey, the question "Did you find other people's comments useful for exploring the data?" received the highest marks (M = 4.46/5.0, SD = 0.63).

5.7. Doubly linked discussions

We also wanted to investigate participant reaction to the doubly linked model of comments. All users understood the model readily and no problems were reported when users wanted to comment on a specific view. The model became more problematic when users wanted to comment on multiple views. In this case, the user had to choose one view as primary, comment on that, and then reference the other views, either indirectly in the text or by creating a link from the bookmark trail. Some users expressed the opinion that creating links was a workable solution, while others wanted to be able to simultaneously compare multiple views for purposes of both analysis and commentary. One important aspect of doubly linked discussions is the problem of determining identical views, despite potentially differing visualization parameters. In this respect, we found our indexing scheme improved the odds of discovering existing commentary while navigating the visualization. Across both lab studies, 28.2% of all unique visits to a visualization state were to a view that had been reached through two or more differing parameter settings. Without the view indexing, there would be a much higher potential for "cross talk," where users post comments concerning similar observations on related views, unaware of each other. Nonetheless, cross talk was observed in a total of six cases, typically when both normalized and absolute axis scales led to similar views. In two cases, participants added linking comments that bridged the related discussions.

5.8. User experience

Overall, users found using sense.us both enjoyable and informative. In the exit survey, the question "Did you enjoy using the system?" received a mean rating of 4.0/5.0 (SD = 0.52). The question "Did you learn something interesting using the system?" received a mean rating of 4.2/5.0 (SD = 0.65). Users also provided usability remarks and suggested additional collaboration features. The next section addresses a number of these requests (Figure 10).

6. DISCUSSION

The usage we observed echoed some of the earlier findings about social data analysis.[27] In particular, we saw cascading conversation threads in which users asked questions, stated hypotheses, and proposed explanations, all in a social context. A significant number of comments were playful or joking, as were a few graphical annotations. It has been hypothesized that one of the spurs to social data analysis is a situation in which each user brought a unique perspective to bear.[27] In the case of job data, this unique perspective was the set of professions of friends and family of the user. We did indeed see people exploring in this fashion, covering a broad set of the data.

On the other hand, we observed a somewhat more businesslike tone to analysis than was seen previously. This was likely in part due to the corporate and laboratory settings of use. The presence of an observer in the lab studies undoubtedly also influenced results, though many users reported they had fun conducting social data analysis.

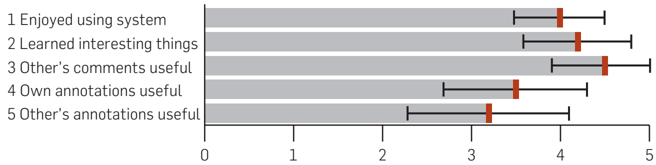
Further research is clearly needed to understand the broad principles of analytical behavior in the context of visualizations. Since the original publication of this article, some of that research has occurred. In the next sections, as we describe research directions suggested by reactions to sense.us, we also provide brief notes on how recent work has shed light on issues of collaboration around visualizations.

6.1. Mechanisms for social data analysis

The doubly linked discussion model was probably the most effective and well-liked novel feature of sense.us. If there was any frustration with this feature, it was that users had to navigate to a precise location to see related comments. This shortcoming, coupled with the high rate of within-applet navigation (Figure 9), raises an intriguing question for future research: would it be helpful to embed social navigation cues in the visualization or interface widgets themselves?

For example, a dynamic query widget used to filter the visualization might include visual cues of how many people have visited or commented on the views reachable using the widget, providing information scent by which the user can purposefully navigate toward either popular or unpopular views. Such widgets could aid the discovery of interesting trends that simply had not yet been seen. In our context, one might imagine a slider—controlling a view parameter—with marks indicating the presence of comments at specific parameter values. Similar techniques can be devised for other interface widgets. A recent system for such "scented widgets"[28] provides evidence that such cues can result in increased revisitation to popular views while also directing users' attention to under-explored data regions.

Figure 10: Results of poststudy survey. Mean values are shown, error bars indicate standard deviation.



A second approach, suggested by many users, would be to show commentary related, though not directly attached to, the current view. Requested features include showing comments from other views that contain links to the current view (*"trackbacks"*), and related commentary on *"nearby"* or *"similar"* views. The latter could help alleviate cross talk. Along these lines, there are appealing possibilities for generalizing the notion of view indexing, for example, suggesting conversations on views deemed semantically similar to the current view. This would require an index of visualization state providing not just equality comparisons, but distance measures. Such a retrieval model might be used to provide additional benefits, such as general searchability and data-aware autocomplete mechanisms.

Users have also suggested using visitation data or explicit ratings of "interestingness" to suggest views of potential interest. Others suggested supporting keyword tagging of comments[22] and mining usage data. For example, both manual and automated tagging of questions or other action items could be used to help direct collaborative effort.

The scope of comment visibility is a larger issue that affects all discussion models. What happens when the amount of discussion becomes untenably large, or users don't want their activity exposed to everyone? The ability to form groups and limit comment visibility to group members is one means requested by users to support privacy and make discussion-following both more relevant and tractable.

Although individual usage varied substantially, most lab study users (87.5%) did use the bookmark trails, which proved essential for comments that included multiple views. Multiple users remarked on the usefulness of the bookmark trails and wanted to more easily share trails as first class objects. At times, users were frustrated when following multiple links in a comment, as the original comment would disappear when a new view was loaded, requiring use of the back button to perform "hub-and-spoke" browsing. In response, users suggested adding a dedicated "presentation" mode to facilitate tours and storytelling.

Finally, the graphical annotations saw significant usage, despite mixed reactions from users. Though they were used for pointing, many users did not find them necessary for disambiguation. We expect that the value of annotations varies significantly depending on the type of visualization being referenced. Regardless, annotations were used regularly for pointing and sometimes for socializing.

If the free-form annotations prove helpful, a second challenge would be to extend them to cover dynamic or evolving data sets. The decoupled nature of geometric annotations can prove problematic when the underlying data changes. Similar problems have been investigated in the context of document annotation.[5] More recent work[19] has explored "data-aware" annotations that translate user selections into declarative queries over the underlying data, allowing annotations to be applied to time-varying data and different visual encodings.

6.2. Communities and data

Since the original sense.us experiment, there have been several new examples of systems that support conversation around data. Web sites such as Swivel.com have provided social-network-style platforms for conversation around data, along with basic charting capabilities. Tableau Software launched its Tableau Server product, which (much like Spotfire's DecisionSite Posters) allows users to collaborate asynchronously around intranet-based visualizations. Little has been published about usage of these systems, however.

One new system where results have been reported is the Many Eyes Web site.[26] In contrast to sense.us, or tools like Tableau or Spotfire, Many Eyes is freely available on the public internet and allows users to upload their own data. Unlike data-oriented sites like Swivel, Many Eyes lets users apply more than a dozen interactive visualization techniques. They may then have discussions around visualizations, though annotation capabilities are more basic than in sense.us. The experiences on the site[26] lend support to the idea that visualization can catalyze discussion. While these discussions can be analytical, they also can be purely social, partisan, or game-like. In addition, the move from a closed setting to the public internet has made clear that these discussions can be highly distributed,[13] with a significant proportion of collaboration occurring (via hyperlinks) off the site. Designing for this type of multisite conversation suggests a whole new set of challenges.

7. CONCLUSION

In this article, we investigated mechanisms supporting asynchronous collaboration around interactive information visualization, seeking to more tightly tie the perceptual and cognitive benefits of visualization to social processes of sensemaking. To do so, we implemented a collaborative data visualization site, sense.us. We then observed usage of the site in order to better understand the social dynamics surrounding collective use of visualizations as well as the efficacy of the particular features.

The features of the site—doubly linked discussions, bookmark trails, geometric annotations, and comment listings—were all exploited by users. The doubly linked discussions successfully enabled users to fluidly transfer attention between visualization and commentary and we suggested ways to further improve this type of discussion. Bookmark trails and geometric annotations were also well used, enabling tours through multiple views and pointing to items of interest, respectively. Finally, users played the roles of both voyager and voyeur, alternating between data-driven exploration directly within the visualization and social navigation through comment listings and user profiles to discover new views of interest.

Overall, we believe these results show the value of focusing on the social aspects of visual analysis. Our user studies indicate that combining conversation and visual data analysis can help people explore a data set both broadly and deeply. From a design perspective, there lies a promising opportunity for exploring new widgets and modes of interaction aimed at enhancing collaboration.

Acknowledgments

We thank Jesse Kriss, Frank van Ham, Doug Fritz, Kate Hollenbach, Steve Whittaker, David Millen, and Maneesh Agrawala for insightful discussions. We also would like to thank all the study participants and users of our system.

References

1. Anupam, V., Bajaj, C.L., Schikore, D., and Shikore, M. Representations in distributed cognitive tasks. *IEEE Computer 27*, 7 (1994), 37–43.
2. Benbunan-Fich, R., Hiltz, S.R., and Turoff, M. A comparative content analysis of face-to-face vs. asynchronous group decision making. *Decision Support Systems 34*, 4 (2003), 457–469.
3. Benko, H., Ishak, E.W., and Feiner, S. Collaborative mixed reality visualization of an archaeological excavation. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2004)* (Arlington, VA, 2004), 132–140.
4. Brodlie, K.W., Duce, D.A., Gallop, J.R., Walton, J.P.R.B., and Wood, J.D. Distributed and collaborative visualization. *Computer Graphics Forum 23*, 2 (2004), 223–251.
5. Brush, A., Bargeron, D., Gupta, A., and Cadiz, J. Robust annotation positioning in digital documents. In *Proceedings of ACM Human Factors in Computing Systems (CHI 2001)*, 2001.
6. Card, S.K., Mackinlay, J.D., and Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan-Kaufmann, 1999.
7. Carter, S., Mankoff, J., and Goddi, P. Building connections among loosely coupled groups: Hebb's rule at work. *Journal of Computer-Supported Cooperative Work 13*, 3 (2004), 305–327.
8. Chuah, M.C. and Roth, S.F. Visualizing common ground. In *Proceedings of the Conference on Information Visualization (IV)* (Los Alamitos, USA, 2003), IEEE Computer Society, 365–372.
9. Chui, Y.-P. and Heng, P.-A. Enhancing view consistency in collaborative medical visualization systems using predictive-based attitude estimation. In *First IEEE International Workshop on Medical Imaging and Augmented Reality (MIAR'01)* (Hong Kong, China), 2001.
10. Churchill, E.F., Trevor, J., Bly, S., Nelson, L., and Cubranic, D. Anchored conversations: chatting in the context of a document. In *ACM CHI 2000*, 2000.
11. Clark, H.H. Pointing and placing. In *Pointing. Where Language, Culture, and Cognition Meet*. S. Kita, ed. Erlbaum, 2003, 243–268.
12. Clark, H.H. and Brennan, S.E. Grounding in communication. In *Perspectives on Socially Shared Cognition (1991)*, American Psychological Association, 127–149.
13. Danis, C.M., Viégas, F.B., Wattenberg, M., and Kriss, J. Your place or mine? Visualization as a community component. In *Proceedings of ACM Human Factors in Computing Systems (CHI 2008)*, 2008.
14. Dietz, P.H. and Leigh, D.L. Diamondtouch: A multi-user touch technology. In *Proceedings of ACM Symposium on User Interface Software and Technology*, 2001, 219–226.
15. Dourish, P. and Chalmers, M. Running out of space: Models of information navigation. In *Human Computer Interaction (HCI'94)*, 1994.
16. Dynamics, G. Command post of the future. Website. Accessed: November, 2007.
17. Grudin, J. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM 37*, 1 (1994), 92–105.
18. Heer, J. Socializing visualization. In *ACM CHI 2006 Workshop on Social Visualization*, 2006.
19. Heer, J., Agrawala, M., and Willett, W. Generalized selection via interactive query relaxation. In *ACM CHI 2008*, 959–969.
20. Hill, W.C. and Hollan, J.D. Deixis and the future of visualization excellence. In *Proceedings of IEEE Visualization* (1991), 314–319.
21. Johansen, R. *Groupware: Computer Support for Business Teams*. The Free Press, New York, 1988.
22. Millen, D.R., Feinberg, J., and Keer, B. Dogear: Social bookmarking in the enterprise. In *Proceedings of ACM CHI 2006*, 111–120.
23. Thomas, J. and Cook, K. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society, 2005.
24. Viégas, F.B., danah boyd, Nguyen, D.H., Potter, J., and Donath, J. Digital artifacts for remembering and storytelling: PostHistory and social network fragments. In *Proceedings of Hawaii International Conference on System Sciences (HICCSS)* (2004), 105–111.
25. Viégas, F.B. and Wattenberg, M. Communication-minded visualization: A call to action. *IBM Systems Journal 45*, 4 (2006), 801–812.
26. Viégas, F.B., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. Many Eyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Nov./Dec. 2007), 1121–1128.
27. Wattenberg, M. and Kriss, J. Designing for social data analysis. *IEEE Transactions on Visualization and Computer Graphics 12*, 4 (July–August 2006), 549–557.
28. Willett, W., Heer, J., and Agrawala, M. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov./Dec. 2007), 1129–1136.

Jeffrey Heer (jheer@stanford.edu), Computer Science Department, Stanford University, Stanford, CA.

Fernanda B. Viégas (viegasf@us.ibm.com),Visual Communication Lab IBM T.J. Watson Research Center, Cambridge, MA.

Martin Wattenberg (mwatten@us.ibm.com), Visual Communication Lab, IBM T.J. Watson Research Center, Cambridge, MA.

CAREERS

Baylor University
Assistant to Full Professor of Computer Science

The Department of Computer Science seeks a productive scholar and dedicated teacher in the area of Game and Simulated Environments for a tenure-track position beginning August, 2009. The ideal candidate will hold a terminal degree in Computer Science or a closely related field, demonstrate scholarly capability in game design and development, and exhibit a passion for teaching and mentoring at the graduate and undergraduate level. For position details and application information please visit: http://cs.baylor.edu/employment

Baylor, the world's largest Baptist university, holds a Carnegie classification as a "high-research" institution. Baylor's mission is to educate men and women for worldwide leadership and service by integrating academic excellence and Christian commitment within a caring community. Baylor is actively recruiting new faculty with a strong commitment to the classroom and an equally strong commitment to discovering new knowledge as Baylor aspires to become a top tier research university while reaffirming and deepening its distinctive Christian mission as described in Baylor 2012 (http://www.baylor.edu/vision/).

Baylor is a Baptist university affiliated with the Baptist General Convention of Texas. As an AA/EEO employer, Baylor encourages minorities, women, veterans, & persons with disabilities to apply.

Baylor University
Assistant to Full Professor of Bioinformatics

The Baylor Department of Computer Science seeks a dynamic scholar to fill this position beginning August, 2009. For complete position details and application information please visit: http://cs.baylor.edu/employment.

Candidates should possess an earned doctorate in Bioinformatics, Computer Science, or a related field, and have sufficient expertise and experience to teach and perform research in the area of Bioinformatics. The successful candidate will be expected to establish, or bring, an active independently-funded research program in areas related to systems biology, genomics, or other areas related to ongoing research within the Department of Computer Science. The successful candidate will also assist in the teaching of graduate and undergraduate computer science and bioinformatics students. Your application should consist of a letter of interest, curriculum vitae, transcripts (as appropriate) and a list of three references. Salary will be commensurate with experience and qualifications.

Baylor, the world's largest Baptist university, holds a Carnegie classification as a "high-research" institution. Baylor's mission is to educate men and women for worldwide leadership and service by integrating academic excellence and Christian commitment within a caring community. Baylor is actively recruiting new faculty with a strong commitment to the classroom and an equally strong commitment to discovering new knowledge as Baylor aspires to become a top tier research university while reaffirming and deepening its distinctive Christian mission as described in Baylor 2012 (http://www.baylor.edu/vision/).

Baylor is a Baptist university affiliated with the Baptist General Convention of Texas. As an AA/EEO employer, Baylor encourages minorities, women, veterans, & persons with disabilities to apply.

Bowling Green State University
Computer Science Assistant Professor

The Department of Computer Science at Bowling Green State University invites applications for a tenure-track faculty position at the Assistant Professor level to begin in August 2009. A Ph.D. in Computer Science or closely related area required. Applicants in all areas of computer science will be considered with preference given to those with expertise in one or more of the following: computer graphics and virtual environments, computer security, distributed systems, human-computer interaction, parallel processing, programming languages, simulation or software engineering. Candidates should present evidence that they are able to teach a wide variety of courses in computer science. Applications must be submitted electronically by January 31, 2009. A detailed description of this position, including application procedures, qualifications, and duties, is available at http://www.bgsu.edu/departments/compsci/jobs/facsearch.html.

Questions may be directed to Faculty Search Committee Chair, at cssearch2009@cs.bgsu.edu or +1 419 372 2337. Bowling Green State University is an affirmative action and equal opportunity employer and educational institution.

Bucknell University
Department of Computer Science
Assistant Professor

Applications are invited for a tenure track entry-level (three or fewer years of full-time teaching experience) assistant professor position in computer science beginning mid-August 2009. Candidates must have completed their Ph.D. requirements in computer science or a closely related field by August 15th, 2009. A strong commitment to excellence in teaching and scholarship is required. The successful candidate must be able to participate in the teaching of required core courses and be able to develop elective courses in the candidate's area of expertise that enhance the department's course offerings.

Bucknell is a highly selective private university emphasizing quality undergraduate education in engineering and in liberal arts and sciences. The B.S. programs in computer science are ABET and CAC accredited. The computing environment is Linux/Unix-based. More information about the department can be found at:

http://www.bucknell.edu/ComputerScience/

Applications will be considered as received and recruiting will continue until the position is filled. Candidates are asked to submit a cover letter, CV, graduate transcript, a statement of teaching philosophy and research interests, and the contact information for three references. Please submit your application to

http://jobs.bucknell.edu/

by searching for the "Computer Science Faculty Position".

Please direct any questions to Professor Xiannong Meng of the Computer Science Department at xmeng@bucknell.edu.

Bucknell University values a diverse college community and is committed to excellence through diversity in its faculty, staff and students. An Equal Opportunity/Affirmative Action Employer, Bucknell University especially welcomes applications from women and minority candidates.

Carnegie Mellon University
Engineering and Public Policy

Engineering and Public Policy at Carnegie Mellon seeks doctoral students with strong technical backgrounds to address policy issues in trustworthy computing and information security; information privacy for Internet applications (social networks, e-commerce, etc.); telecommunication, Internet and spectrum policy; electric energy systems; risk analysis and regulation. Opportunities for dual-degrees with Portugal available. See http://www.epp.cmu.edu. Victoria Finney, EPP, Carnegie Mellon, Pittsburgh, PA 15213 USA

Columbia University
Department of Computer Science

The Department of Computer Science is seeking applicants for a tenure-track position at either the junior or senior level in computer graphics. Applicants should have a Ph.D. in a relevant field, and have demonstrated excellence in research and the potential for leadership in the field. Senior applicants should also have demonstrated excellence in teaching and continued strong leadership in research.

Our department of 34 tenure-track faculty and 1 lecturer attracts excellent Ph.D. students, virtually all of whom are fully supported by research grants. The department has active ties with major industry partners including Adobe, Autodesk, Disney, Dreamworks, Nvidia, Sony, Weta and also to the nearby research laboratories of AT&T, Google, IBM(T.J. Watson), NEC, Siemens, Telcordia

Technologies and Verizon. Columbia University is one of the leading research universities in the United States, and New York City is one of the cultural, financial, and communications capitals of the world. Columbia's tree-lined campus is located in Morningside Heights on the Upper West Side.

Applicants should submit a CV, summaries of research and teaching interests, and the names and email addresses of at least three references to: academicjobs.columbia.edu/applicants/Central?quickFind=51000.

The search will remain open until filled, but will close no sooner than Feb 28, 2009.

Duke University
Department of Computer Science

The Department of Computer Science at Duke University invites applications and nominations for faculty positions at all levels, to begin August 2009. We are interested in strong candidates in all active research areas of computer science, both core and interdisciplinary areas, including algorithms, artificial intelligence, computational biology, computational economics, computer architecture, computer vision, database systems, distributed systems, machine learning, networking, and security.

The department is committed to increasing the diversity of its faculty, and we strongly encourage applications from women and minority candidates.

A successful candidate must have a solid disciplinary foundation and demonstrate promise of outstanding scholarship in every respect, including research and teaching. Please refer to www.cs.duke.edu for information about the department.

Applications should be submitted online at www.cs.duke.edu/facsearch. A Ph.D. in computer science or related area is required. To guarantee full consideration, applications and letters of reference should be received by January 4, 2009.

Durham, Chapel Hill, and the Research Triangle of North Carolina are vibrant, diverse, and thriving communities, frequently ranked among the best places in the country to live and work. Duke and the many other universities in the area offer a wealth of education and employment opportunities for spouses and families.

Duke University
Faculty Position
Department of Electrical and Computer Engineering

The Pratt School of Engineering invites applications for a tenure track or tenured faculty position in the Department of Electrical and Computer Engineering. The department is primarily seeking candidates in Computer Engineering and/or Communications who have demonstrated a strong commitment to excellence in both scholarly research and innovative teaching. Successful candidates must demonstrate an ability to develop a world-class research program, be capable of obtaining competitive external research funding, and actively participate in and be committed to outstanding teaching at both the undergraduate and graduate levels.

Applicants should submit a cover letter (including email, address, and phone number) and research and teaching statements describing

UNIVERSITÄT DES SAARLANDES

M2CI CLUSTER OF EXCELLENCE

Saarland University is seeking to establish several

Junior Research Groups (W1/W2)

within the recently established Cluster of Excellence "Multimodal Computing and Interaction" which was established by the German Research Foundation (DFG) within the framework of the German Excellence Initiative.

The term "multimodal" describes the different types of digital information such as text, speech, images, video, graphics, and high-dimensional data, and the way it is perceived and communicated, particularly through vision, hearing, and human expression. The challenge is now to organize, understand, and search this multimodal information in a robust, efficient and intelligent way, and to create dependable systems that allow natural and intuitive multimodal interaction. We are looking for highly motivated young researchers with a background in the research areas of the cluster, including algorithmic foundations, secure and autonomous networked systems, open science web, information processing in the life sciences, visual computing, large-scale virtual environments, synthetic virtual characters, text and speech processing and multimodal dialog systems. Additional information on the Cluster of Excellence is available on http://www.mmci.uni-saarland.de. Group leaders will receive junior faculty status at Saarland University, including the right to supervise Bachelor, Master and PhD students. Positions are limited to five years.

Applicants for W1 positions (phase I of the program) must have completed an outstanding PhD. Upon successful evaluation after two years, W1 group leaders are eligible for promotion to W2. Direct applicants for W2 positions (phase II of the program) must have completed a postdoc stay and must have demonstrated outstanding research potential and the ability to successfully lead their own research group. Junior research groups are equipped with a budget of 80k to 100k Euros per year to cover research personnel and other costs.

Saarland University has leading departments in computer science and computational linguistics, with more than 200 PhD students working on topics related to the cluster (see http://www.informatik-saarland.de for additional information). The German Excellence Initiative recently awarded multi-million grants to the Cluster of Excellence "Multimodal Computing and Interaction" as well as to the "Saarbrücken Graduate School of Computer Science". An important factor to this success were the close ties to the Max Planck Institute for Computer Science, the German Research Center for Artificial Intelligence (DFKI), and the Max Planck Institute for Software Systems which are co-located on the same campus.

Candidates should submit their application (curriculum vitae, photograph, list of publications, short research plan, copies of degree certificates, copies of the five most important publications, list of five references) to the coordinator of the cluster, Prof. Hans-Peter Seidel, MPI for Computer Science, Campus E1 4, 66123 Saarbrücken, Germany. Please, also send your application as a single PDF file to applications@mmci.uni-saarland.de.

The review of applications will begin on January 15, 2009, and applicants are strongly encouraged to submit applications by that date; however, applications will continue to be accepted until January 31, 2009. Final decisions will be made following a candidate symposium that will be held during March 9 – 13, 2009.

Saarland University is an equal opportunity employer. In accordance with its policy of increasing the proportion of women in this type of employment, the University actively encourages applications from women. For candidates with equal qualification, preference will be given to people with physical disabilities.

their interests and qualifications, curriculum vitae along with the names and addresses of five references, and copies of three publications to ecurrin@ee.duke.edu. Applications received before February 1, 2009 will be given full consideration. Duke University is an Equal Opportunity Employer. Candidates from underrepresented groups in ECE are particularly encouraged to apply. Additional information about Duke's ECE department is available online at http://www.ee.duke.edu.

D.E. Shaw India Software Private Limited
Software Developer

The D. E. Shaw Group, a global investment and technology development firm with approximately $36 billion in aggregate capital, seeks exceptional software developers to build trading systems, real-time data feed infrastructure, reconcilers, workflow systems, etc.

Candidates should have an outstanding academic record and a sound understanding of computer systems, programming paradigms, and software engineering practices. Experience in building applications used in the financial markets would be useful, but we encourage candidates who have built non-trivial systems of any type to apply.

We provide unusual opportunities for growth, and we compensate extraordinary people extraordinarily well. To apply, e-mail your resume to dev-india@recruit.deshaw.com. EOE.

Georgetown University
Senior Faculty Position
Professor and Department Chair
Department of Computer Science

The Department of Computer Science at Georgetown University seeks a dynamic scholar/educator for the position of Professor and Department Chair. Georgetown University is entering a phase of substantial growth in science, during which the successful candidate will be expected to shape the direction of computer science at Georgetown, through the development of new programs/initiatives and the expansion of collaborations across the University. The individual selected must have an international reputation as a scholar, experience as a successful educator, and demonstrated leadership ability. Subject to review by the University Committee on Rank and Tenure, this position will be a tenured appointment at the Full Professor level.

Currently, the department has degree programs at the B.S. and M.S. level, and consists of 7 tenured/tenure-track faculty members, 5 adjunct faculty members, and a full-time administrative coordinator. Review of applications and nominations will begin December 15, 2008, and will continue until the position is filled. Candidates from all areas and sub-disciplines of computer science and related areas are encouraged to apply. The current research interests of the faculty are in algorithms, artificial intelligence, database systems, data mining, nonstandard computing, security, and software engineering. Please visit the department's website for additional information: http://www.cs.georgetown.edu/.

Only electronic applications will be accepted. Please send a cover letter, curriculum vitae, research/teaching statements, and the names of 5 references, as a single pdf file, to: deptchairsearch@cs.georgetown.edu.

Georgetown University is an Equal Opportunity/Affirmative Action Employer. We are committed to creating an environment that values and supports diversity, equity and inclusiveness across our campus community and encourage applications from qualified individuals who will help us achieve this mission.

Gonzaga University
Assistant or Associate Professor of Computer Science

Gonzaga University seeks applicants for an Assistant or Associate Professor of Computer Science. This is a full-time, tenure track position to begin in the fall semester, 2009. Required qualifications: demonstrated expertise in data mining, database management systems, scientific visualization, or bioinformatics; Ph.D. in computer science or closely related field. The Department's facilities include a computational science lab with a 512 node cluster, and a sensor network and robotics lab. The department is housed in the newly completed Paccar Center for Applied Science.

Gonzaga, with 7000 students, is in the center of Spokane, Washington along the Spokane River. Research opportunities are available with the Pacific Northwest National Laboratories and many businesses in the area. Spokane, the health care

Assistant Professor Positions, Department of Computer Science

MO SHUILE TOGAM SUAS

UNIVERSITY OF CALGARY

The Department of Computer Science at the University of Calgary seeks outstanding candidates for several tenure-track positions at the Assistant Professor level. Applicants from database management, software engineering, information security, theory, HCI/Information Visualization, and computer games are of particular interest. Details for each position appear at www.cpsc.ucalgary.ca/career. Applicants must possess a doctorate in Computer Science or a related discipline at the time of appointment, and have strong potential to develop an excellent research record.

The Department is one of Canada's leaders as evidenced by our commitment to excellence in research and teaching. It has an expansive graduate program and extensive state-of-the-art computing facilities. Calgary is a multicultural city that is a fast-growing city in Canada. Calgary enjoys a moderate climate located beside the natural beauty of the Rocky Mountains.

Further information about the Department is available at http://www.cpsc.ucalgary.ca/. Interested applicants should send a CV, a concise description of their research area and program, a statement of teaching philosophy, and arrange to have at least three reference letters sent to: Dr. Frank Maurer, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada, T2N 1N4 or via email to: search@cpsc.ucalgary.ca. The applications will be reviewed beginning November 2008 and continue until the positions are filled.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

Research Opportunities with Disney Research, part of The Walt Disney Company

Senior Research Scientists, Research Scientists, and Postdocs

Disney Research, with research labs in Pittsburgh, Los Angeles, Zurich, Salt Lake City, and San Francisco, invites applications from outstanding researchers at all levels in the areas of:

AI & Autonomous Agents • Audio Processing • Computer Graphics
Computer & Sensor Networks • Computer Vision
Cryptography & Information Security • Display Technologies
Distributed & Parallel Computing • Embedded Systems
Human-Computer Interaction • Machine Learning & Data Mining
Materials Science • Mobile Computing
Multimedia Content Management • Operations Research
Radios & Antennas • Robotics • Speech Recognition • Video Processing

Candidates should have a strong record of achievement in applied research, excellent academic credentials, and an earned Ph.D. in their area of specialization. Work experience in the entertainment industry is preferred but not required.

For more information and to apply, visit
DisneyResearch.com

Disney Research

EOE • Drawing Creativity from Diversity • ©Disney center for the inland Northwest, has a metropolitan area population of 500,000. The city offers one of the finest four-season living environments in the Pacific Northwest, with five ski resorts, more than 60 lakes, and several national forests nearby.

Review of applications will begin 1/12/09. Applications will be accepted until the position is filled. Please send a letter, complete curriculum vita, a statement of research and teaching objectives, and the names, addresses, and telephone numbers of at least three references to: Paul De Palma, Chair, Department of Computer Science, Gonzaga University, Spokane, WA 99258-0026. Electronic submissions in pdf format are preferred and should be sent to: depalma@gonzaga.edu. Gonzaga is a Catholic, Jesuit and humanistic university interested in candidates who can contribute to its distinctive mission. The University is an AA/EEO employer and educator committed to diversity.

Helsinki Institute for Information Technology (HIIT)
Principal Scientist

Helsinki Institute for Information Technology HIIT invites applications for a fixed-term position of PRINCIPAL SCIENTIST for three years starting on 1 January 2009. Closing date for applications 10 November 2008. More info at www.hiit.fi/jobs.

Hong Kong Polytechnic University
Department of Computing

The Department invites applications for Professors/Associate Professors/Assistant Professors in Database and Information Systems / Biometrics, Computer Graphics and Multimedia / Software Engineering and Systems / Networking, Parallel and Distributed Systems. Applicants should have a PhD degree in Computing or closely related fields, a strong commitment to excellence in teaching and research as well as a good research publication record. Applicants with extensive experience and a high level of achievement may be considered for the post of Professor/Associate Professor.

Please visit the website at http://www.comp.polyu.edu.hk/ for more information about the Department. Salary offered will be commensurate with qualifications and experience. Initial appointments will be made on a fixed-term gratuity-bearing contract. Re-engagement thereafter is subject to mutual agreement. Remuneration package will be highly competitive.

Applicants should state their current and expected salary in the application. Please submit your application via email to hrstaff@polyu.edu.hk. Application forms can be downloaded from http://www.polyu.edu.hk/hro/job.htm.

Recruitment will continue until the positions are filled. Details of the University's Personal Information Collection Statement for recruitment can be found at http://www.polyu.edu.hk/hro/jobpics.htm.

Indiana University of Pennsylvania
Computer Science Department

Tenure-track position at the Assistant/Associate Professor level. For details, call 724-357-7994 or http://www.iup.edu/employment. IUP is an equal opportunity employer M/F/H/V.

KAUST

King Abdullah University of Science and Technology (KAUST)

Faculty Openings in Computer Science and Applied Mathematics

King Abdullah University of Science and Technology (KAUST) is being established in Saudi Arabia as an international graduate-level research university dedicated to inspiring a new age of scientific achievement that will benefit the region and the world. As an independent and merit-based institution and one of the best endowed universities in the world, KAUST intends to become a major new contributor to the global network of collaborative research. It will enable researchers from around the globe to work together to solve challenging scientific and technological problems. The admission of students, the appointment, promotion and retention of faculty and staff, and all the educational, administrative and other activities of the University shall be conducted on the basis of equality, without regard to race, color, religion or gender.

KAUST is located on the Red Sea at Thuwal (80km north of Jeddah). Opening in September 2009, KAUST welcomes exceptional researchers, faculty and students from around the world. To be competitive, KAUST will offer very attractive base salaries and a wide range of benefits. Further information about KAUST can be found at http://www.kaust.edu.sa/.

KAUST invites applications for faculty position at all ranks (Assistant, Associate, Full) in Applied Mathematics (with domain applications in the modeling of biological, physical, engineering, and financial systems) and Computer Science, including areas such as Computational Mathematics, High-Performance Scientific Computing, Operations Research, Optimization, Probability, Statistics, Computer Systems, Software Engineering, Algorithms and Computing Theory, Artificial Intelligence, Graphics, Databases, Human-Computer Interaction, Computer Vision and Perception, Robotics, and Bio-Informatics (this list is not exhaustive). KAUST is also interested in applicants doing research at the interface of Computer Science and Applied Mathematics with other science and engineering disciplines. High priority will be given to the overall originality and promise of the candidate's work rather than the candidate's sub-area of specialization within Applied Mathematics and Computer Science.

An earned Ph.D. in Applied Mathematics, Computer Science, Computational Mathematics, Computational Science and Engineering, Operations Research, Statistics, or a related field, evidence of the ability to pursue a program of research, and a strong commitment to graduate teaching are required. A successful candidate will be expected to teach courses at the graduate level and to build and lead a team of graduate students in Master's and Ph.D. research.

Applications should include a curriculum vita, brief statements of research and teaching interests, and the names of at least 3 references for an Assistant Professor position, 6 references for an Associate Professor position, and 9 references for a Full Professor position. Candidates are requested to ask references to send their letters directly to the search committee. Applications and letters should be sent via electronic mail to kaust-search@cs.stanford.edu. The review of applications will begin immediately, and applicants are strongly encouraged to submit applications as soon as possible; however, applications will continue to be accepted until December 2009, or all 10 available positions have been filled.

In 2008 and 2009, as part of an Academic Excellence Alliance agreement between KAUST and Stanford University, the KAUST faculty search committee consisting of professors from the Computer Science Department and the Institute of Computational and Mathematical Engineering at Stanford University, will evaluate applicants for the faculty positions at KAUST. However, KAUST will be responsible for all hiring decisions, appointment offers, recruiting, and explanations of employment benefits. The recruited faculty will be employed by KAUST, not by Stanford. Faculty members in Applied Mathematics and Computer Science recruited by KAUST before September 2009 will be hosted at Stanford University as Visiting Fellows until KAUST opens in September 2009.

Open Position at INRIA:

Technology Development Director

INRIA is a French public research institute in information and communication science and technology. The institute has about 160 project-teams throughout eight research centres in partnerships with universities and other research organization. INRIA focuses the activity of over 3000 scientists on research, development and industry transfer activities in the following computer science and applied mathematics areas:

- Modeling, simulation and optimization of complex dynamic systems
- Formal methods in programming secure and reliable computing systems
- Networks and ubiquitous information, computation and communication systems
- Vision and human-computer interaction modalities, virtual worlds and robotics
- Computational Engineering, Computational Sciences and Computational Medicine

INRIA is seeking applications for the position of Technology Development (TD) Director, one of the five scientific directors of INRIA, a position located at INRIA headquarters near Paris.

The Director has to develop and manage the technology development activities of INRIA, to supervise all software and technology development activities, coordinate the support units for TD in the eight research centers and organize the technology development platforms. He or she implements the partnership policy for R&D in the field of software and technology development.

Applications are sought from individuals having the experience and proficiency required for accomplishing the above duties with:

- A very good knowledge of research and development in ICST
- Accomplishments in and a wide experience of software and technology development
- Abilities to manage, lead, and interact with all researchers and engineers.

The TD Director position corresponds to an attractive salary package. It is to be filled at the beginning of 2009 (starting date negotiable). Applications and enquiries should be sent to INRIA's Chairman no later than January 15, 2009 (or at: http://www.inria.fr/index.en.html).

Iowa State University of Science & Technology

Announcement of Lanh and Oanh Nguyen Endowed Chair in Software Engineering

The Department of Computer Science at Iowa State University invites applications from distinguished scholars for the new **Lanh and Oanh Nguyen Endowed Chair in Software Engineering** beginning August, 2009. We are especially interested in candidates whose research integrates software engineering with other areas of computer science or interdisciplinary applications. The chair-holder will provide leadership in conducting research in advanced and emerging technologies, engage with top research institutions, stimulate transfer of innovative technologies to the private and public sectors, and enhance student learning in the recently established software engineering B.S. program.

A Ph.D. or equivalent in computer science, software engineering, or a closely related field is required. The successful candidate must have a nationally and internationally recognized record of outstanding research and publication in software engineering as well as demonstrated leadership in instruction and in expanding funded research programs. It is anticipated that the chair-holder will have the credentials to join the department as a full professor.

Our department consists of 27 full-time tenure-track faculty members with a strong core of research-active software engineering faculty. We offer B.S., M.S., and Ph.D. degrees in Computer Science and participate in new B.S. degrees in Software Engineering and in Bioinformatics and Computational Biology. We also participate in interdepartmental graduate programs in Bioinformatics and Computational Biology, Human-Computer Interactions, and Information Assurance. We have strong research and educational programs in Algorithms and Complexity, Artificial Intelligence, Bioinformatics and Computational Biology, Databases, Data Mining, Information Assurance, Programming Languages, Multimedia Systems, Operating Systems and Networks, Robotics, and Software Engineering. Our department has over $4.2 million in active research grants. Including the above interdisciplinary activities, we contribute to active research and training grants totaling approximately $20 million.

Applicants should send via email a curriculum vita including teaching and research statements, and the names and email addresses of at least five references, to endowedchair-search@cs.iastate.edu.

Chair of Search Committee
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1041
Fax: 515-294-0258
Tel: 515-294-4377
E-mail: endowedchair-search@cs.iastate.edu
Web: www.cs.iastate.edu

Review of applications will begin on January 15, 2009 and will continue until the position is filled.

For more information, please visit us at http://www.cs.iastate.edu

Iowa State University is an equal opportunity employer. Women and members of underrepresented minorities are strongly encouraged to apply. Iowa State University is responsive to the needs of dual-career couples, offers family-friendly policies, and is the recipient of a National Science Foundation ADVANCE award for gender equity.

NOVA SOUTHEASTERN UNIVERSITY

YOUR FUTURE. YOUR TERMS.

Computer Science/Engineering Faculty

The Farquhar College of Arts and Sciences seeks applications for the following faculty positions. Unless specifically noted otherwise, positions are full-time (9.5 month) and entail teaching the equivalent of 8 courses, participation in department and college committees and curriculum development, and strong commitment to student and community service and scholarship.

Candidates with a terminal degree are strongly preferred; faculty rank is dependent on terminal degree, college level teaching experience, and scholarship. Candidates with college-level teaching experience and a plan for active scholarship are preferred. Interest in or experience with teaching or developing online courses is a plus. Review of applications begins immediately; positions begin August 2009.

Duties include the instruction of Computer Sciences courses. Candidates should exhibit a strong commitment to undergraduate teaching, undergraduate program development in computer sciences, computer information systems, and engineering technology programs. Professional/Industry experience is a plus. Master's degree in relevant field required; terminal degree preferred.

With more than 120 full-time faculty members, 250 part time faculty members, and 2600 students, the Farquhar College of Arts and Sciences at Nova Southeastern University offers 21 undergraduate majors, 35 minors, and 5 certificate programs across four academic divisions, as well as a graduate program in writing. The College serves more than 5600 undergraduate students through general education, honors, and study abroad programs. Nova Southeastern University is the sixth largest not-for-profit university in the United States, serving more than 27,000 students. The university is located on 300 acres in Fort Lauderdale, Florida.

For confidential consideration, please apply to position #994322 at **www.nsujobs.com** and submit a letter of interest, resume, and curriculum vitae.

Visit our website: **www.nova.edu**

Nova Southeastern University is an Equal Opportunity/Affirmative Action Employer.

Iowa State University of Science & Technology

Announcement of Opening for Tenure-track Position

The Department of Computer Science at Iowa State University is seeking outstanding candidates to fill a tenure-track position, to commence in August, 2009. We are especially interested in applicants at the assistant professor level in Software Engineering. Successful candidates will have demonstrated potential for outstanding research and instruction in computer science and software engineering. A Ph.D. or equivalent in computer science, software engineering, or a closely related field is required.

Our department currently consists of 27 full-time tenure-track faculty members. We offer B.S., M.S., and Ph.D. degrees in Computer Science and participate in new B.S. degrees in Software Engineering and in Bioinformatics and Computational Biology. We are active in interdepartmental graduate programs in Bioinformatics and Computational Biology, Human-Computer Interactions, and Information Assurance. We have strong research and educational programs in Algorithms and Complexity, Artificial Intelligence, Bioinformatics and Computational Biology, Databases, Data Mining, Information Assurance, Programming Languages, Multimedia Systems, Operating Systems and Networks, Robotics, and Software Engineering. Our department has over $4.2 million in active research grants. Including the above interdisciplinary activities, we contribute to active research and training grants totaling approximately $20 million.

A dynamic faculty, a moderate teaching load (typically 3 courses per year with one course reduction for active researchers and possible further reductions for junior faculty), a strong graduate program, and a well-funded research program provide an excellent academic environment.

Applicants should send via email a curriculum vita including teaching and research statements, and the names and email addresses of at least three references, to faculty-search@cs.iastate.edu.

Chair of Search Committee
Department of Computer Science
Iowa State University
Ames, Iowa 50011-1041
Fax: 515-294-0258
Tel: 515-294-4377
E-mail: faculty-search@cs.iastate.edu
Web: www.cs.iastate.edu

Review of applications will begin on January 15, 2009 and will continue until the position is filled.

Iowa State University is an equal opportunity employer. Women and members of underrepresented minorities are strongly encouraged to apply. Iowa State University is responsive to the needs of dual-career couples, offers family-friendly policies, and is the recipient of a National Science Foundation ADVANCE award for gender equity.

The Purdue School of Engineering & Technology at IUPUI
Tenure-Track Assistant Professor

The Purdue School of Engineering & Technology at IUPUI invites applications for a tenure-track Assistant Professor in Computer & Information Technology to start August 2009. Ph.D. and networking/security expertise required.

Seehttp://www.iupui.edu/˜oeo/academicjobs/IN-ENGT08010htm.HTM for full position description and application process. IUPUI is an AA/EO employer.

Marist College
Assistant Professor of Computer Science

The School of Computer Science and Mathematics at Marist College invites applications for a tenure-track position of Assistant Professor of Computer Science beginning Fall 2009. Applicants must have Ph.D. in Computer Science or closely related discipline by the time of appointment.

To learn more or to apply, please visit https://jobs.marist.edu/ . On-line applications accepted only. AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER

Marist College
Assistant Professor of Information Technology and Systems

The School of Computer Science and Mathematics at Marist College invites applications for a tenure-track position of Assistant Professor of Information Technology and Systems.

Candidates must have a doctoral degree in MIS or closely related field. ABDs may be considered. Excellence in teaching and scholarly work is required. Industry and/ or consulting experience is highly desirable.

To learn more or to apply, please visit https://jobs.marist.edu/ . Online applications accepted only. AN EQUAL OPPORTUNITY/AFFIRMATIVE ACTION EMPLOYER

Memorial University of Newfoundland
Tenure-Track Faculty Position in Bioinformatics

The Departments of Computer Science and Biology at MUN invite applications for a tenure-track

Oregon State UNIVERSITY OSU

School of Electrical Engineering and Computer Science

The School of Electrical Engineering and Computer Science at Oregon State University invites applications for tenure-track positions in Computer Science. The School of EECS strongly encourages teamwork and collaboration within the School, and with other departments and universities. We are particularly interested in candidates who can collaborate with our Graphics/Visualization, End-User Software Engineering, and Machine Learning groups. We seek applications from candidates in the areas of Software Engineering, Human Computer Interaction, or Theoretical Computer Science. Exceptional candidates in other areas will also be considered.

Applicants should have an earned doctorate in Computer Science/Computer Engineering by the appointment start date and demonstrate a strong commitment to high-quality undergraduate and graduate teaching and the development of a vibrant research program.

OSU is one of only two American universities to hold the Land Grant, Sea Grant, Sun Grant, and Space Grant designation and is the only Oregon institution recognized for its "very high research activity" (RU/VH) by the Carnegie Foundation for the Advancement of Teaching. With a faculty of 45, the School of EECS enrolls 1300 undergraduate and 300 MS/PhD students.

For more information, including instructions for application, visit http://www.eecs.oregonstate.edu. OSU is and AA/EOE.

MichiganTech

Endowed Professorships and Faculty Positions in Computational Discovery and Innovation

Michigan Technological University announces a Strategic Faculty Hiring Initiative (SFHI) that will add ten tenure-track positions, including at least two endowed professorships, during 2009. SFHI is an ongoing commitment to substantially expand Michigan Tech's faculty resources in targeted strategic areas of multidisciplinary research and inquiry. This initiative follows on last year's first SFHI which resulted in ten hires in the area of Sustainability.

Michigan Tech seeks to attract exceptional candidates whose interests and capabilities match the following objectives and activities: develop computationally-based tools, processes, and environments; extend the boundaries of high-performance computing (HPC); investigate and model complex systems; foster synergies in research methodologies, computational techniques, and innovation.

Michigan Tech seeks a diverse applicant pool from a wide range of disciplines in this strategic initiative. For full consideration, applications should be received by January 30, 2009; review will continue until all positions are filled. Attractive salary, benefit and start-up packages will be provided for successful applicants.

Details about the Michigan Tech Strategic Faculty Hiring Initiative in Computational Discovery and Innovation are available at www.mtu.edu/sfhi. Applicants should prepare their materials as a single PDF document, and send it as an e-mail attachment to provost@mtu.edu. More general information on Michigan Technological University is available at www.mtu.edu.

Michigan Tech is an internationally renowned doctoral research university located in Michigan's scenic Upper Peninsula, on the south shore of Lake Superior. Houghton provides a unique setting where natural beauty, culture, education, and a diversity of residents from around the world come together to share a superb living and learning experience.

Michigan Technological University is an equal opportunity, affirmative action employer/educational institution. Applications from women and minorities are encouraged.

position in Bioinformatics starting no later than September 1, 2009. Appointment will be at the Assistant Professor level, with primary appointment in Computer Science, and equal responsibility in both departments.

A Ph.D. in Computer Science, Computational Science, Biology, or related fields is required and post-doctoral or equivalent experience is desirable. Applicants should have experience in Bioinformatics, and be keen to do interdisciplinary work between the departments. Applicants should possess a strong research record with outstanding promise for future research, and be able to demonstrate the potential for excellent undergraduate and graduate teaching in Bioinformatics.

Memorial University is committed to employment equity and encourages applications from qualified women and men, visible minorities, aboriginal people and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

Closing date for applications will be January 15, 2009. Details and information on the application procedure may be found at http://www.mun.ca/computerscience/employment/ (Reference # VPA-COSC-2007-001).

North Carolina State University (NCSU)
Teaching Faculty Positions

The Department of Computer Science at North Carolina State University (NCSU) invites applications from outstanding committed teachers for Teaching Assistant Professor position(s) and Lecturer position(s) starting August 2009.

Candidates for Teaching Assistant Professor must have an earned doctorate in an appropriate discipline by August 15, 2009. The ideal candidate for this renewable, non-tenure-track position is someone whose career goals are focused on the development and delivery of excellent and innovative undergraduate Computer Science education. Successful candidates will interact with the regional and national community, and must be student-centered with excellent communication skills. They will also be expected to contribute to department efforts in scholarship and service. Participation in high quality research activities centered on teaching, learning and Computer Science related pedagogy is also expected.

Candidates for Lecturer must have earned at least an M.S. in Computer Science or closely related discipline by August 15, 2009, and have at least two years teaching experience. Primary responsibilities will include teaching undergraduate Computer Science courses, with initial focus on lower-division courses; development of course materials; and management of introductory course sequences – including supervision of Teaching Assistants and development of curricular materials, classroom demonstrations, and laboratory exercises.

For more details on the department and its programs, the university, and the area, see http://www.csc.ncsu.edu/.

Applications will be reviewed as they are received. The positions will remain open until suitable candidates are identified, although applicants are encouraged to apply by December 31, 2008. Applicants for either position should submit cover letter, curriculum vitae, teaching statement, research and/or service statement, and names and complete contact information of four references, including email addresses and phone numbers. Additional materials, especially evidence of teaching quality and innovation, are encouraged. Application materials should be submitted online in PDF form to http://jobs.ncsu.edu (reference either position number 04 69 0810 for Teaching Assistant Professor or 04 69 0811 for Lecturer). Candidates can obtain more detail about the positions advertised here at http://www.csc.ncsu.edu/employment. Inquiries may be sent via email to teaching_faculty_search@csc.ncsu.edu.

North Carolina State University is an equal opportunity and affirmative action employer. In addition, NC State University welcomes all persons without regard to sexual orientation. Individuals with disabilities desiring accommodations in the application process should contact the Department of Computer Science at (919) 515-2858.

Northern Michigan University
Mathematics / Computer Science / Mathematics Education

Position Type: Department Head, Tenured, Full Professor
Department: Mathematics and Computer Science
Description/Requirements: Visit HigherEdJobs.Com or call (906) 227-2020
Annual Salary: Competitive
Application Deadline: Screening will begin December 15, 2008, and continue until the position is filled. NMU is an AA/EOE.

Northwestern University
Department of Electrical Engineering and Computer Science
Faculty Opening in Computer Engineering and Systems at All Professorial Ranks

The Department of Electrical Engineering and Computer Science at Northwestern University invites applications for a tenure track faculty position in computer engineering and systems. Candidates at all levels will be considered. An earned Ph.D. in Computer Science, Computer Engineering, or a related field is required, as is demonstrated success within computer engineering research and/or experimental computer systems research. More specific areas of interest include embedded systems, parallel systems, high performance computing, distributed systems, operating systems, compilers, power aware systems and sensor networks. Successful candidates will be expected to carry out world class research, collaborate with other faculty, and teach effectively at the undergraduate and graduate levels. Compensation and start-up packages are negotiable and will be competitive.

Northwestern EECS consists of over 50 faculty members of national prominence whose interests span a wide range. The Computer Engineering and Systems division, with 17 faculty members, is likely to provide a particularly stimulating collaborative environment for successful candidates. Northwestern University is located in the Chicago area.

Applicants should send a curriculum vitae, statements of research and teaching interests, three representative papers, and the names of at least three references for junior applicants and five for senior applicants, by email to: ces-search@eecs.northwestern.edu or by regular mail to:

CES Faculty Search Committee
Department of Electrical Engineering and Computer Science
Northwestern University
2145 Sheridan Road
Evanston, IL 60208

To ensure full consideration, applications should be received by January 15, 2009. Preference will be given to early applications, and interviews may start early, but no offer will be made prior to March 2009. The selected applicant will begin in the position either in September, 2009 or January, 2010.

Further information about the hiring department and the University is available at http:/www.eecs.northwestern.edu and http://www.northwestern.edu .

Northwestern University is an Affirmative Action, Equal Opportunity Employer. Women and minorities are encouraged to apply. Hiring is contingent upon eligibility to work in the United States.

Prairie View A&M University
Department of Computer Science
Job Vacancy Number 09-05

The Department of Computer Science, within the College of Engineering at Prairie View A&M University, invites applications for two tenure-track faculty positions at the Assistant Professor level beginning Fall Semester 2009. Candidates with research interests in any area of computer science will be considered. The successful candidate will be required to teach undergraduate and master's level courses in computer science and computer information systems; advise students; provide leadership in research, program/curriculum development; and serve on academic and administrative committees. The Department offers an ABET accredited undergraduate major in Computer Science and Masters degree programs in Computer Science and Computer Information Systems. For more information, visit the web site at http://www.pvamu.edu/cs.

Candidates must have completed a Ph.D. in Computer Science or a related discipline such as Computer Information Systems, Information Technology, or Computer Engineering by the date of the appointment and demonstrate a commitment to excellence in teaching, research, and service. The successful candidate must have knowledge of the education and research process in a United States university setting and use their abilities to effectively develop/teach outcome-based graduate and undergraduate courses and to establish funded research programs.

Applicants should send in ONE packet, in HARD COPY format, a cover letter of application stating the vacancy number, a detailed vitae, a statement of research and teaching interests, the names and complete contact information of three references, and copies of all transcripts to: Chair of the Search Committee, Department of Computer Science, PO Box 519, MS 2515, Prairie View, TX 77446. Incomplete application packets will not be considered. Email submissions will NOT be accepted. All degrees must be from an accredited college or university. If the educational transcript is from a foreign institution, a professional agency evaluation from an approved agency list is required to verify United States' educational equivalence. Prairie View A&M University, a member of the Texas A&M University System, is located approximately 30 minutes west of Houston, TX. The University is an Equal Opportunity Employer. Applications will be accepted until January 30, 2009. Application review will begin immediately and continue until closing date.

Purdue University
Department of Computer Science
Assistant Professor

The Department of Computer Science at Purdue University invites applications for tenure-track positions at the assistant professor level beginning August 2009. Outstanding candidates in all areas of Computer Science will be considered.

The Department of Computer Science offers a stimulating and nurturing academic environment. Forty-four faculty members direct research programs in analysis of algorithms, bioinformatics, databases, distributed and parallel computing, graphics and visualization, information security, machine learning, networking, programming languages and compilers, scientific computing, and software engineering. The department has implemented a strategic plan for future growth supported by the higher administration and recently moved into a new building. Further information about the department is available at http://www.cs.purdue.edu.

All applicants should hold a PhD in Computer Science, or a closely related discipline, be committed to excellence in teaching, and have demonstrated potential for excellence in research. Salary and benefits are highly competitive. Applicants should apply online at https://hiring.science.purdue.edu. Review of applications will begin on October 1, 2008, and will continue until the positions are filled. Purdue University is an Equal Opportunity/Equal Access/Affirmative Action employer fully committed to achieving a diverse workforce.

Queens College of the City University of New York
Assistant Professor of Computer Science (Tenure-Track)

The Department of Computer Science at Queens College of CUNY is accepting applications for a tenure-track position at the Assistant Professor level starting Fall 2009, in any area of computer science. Consult http://www.cs.qc.cuny.edu/ for further information.

Rose-Hulman Institute of Technology
Visiting Assistant Professor of Computer Science and Software Engineering

The Department of Computer Science and Software Engineering at Rose-Hulman Institute of Technology invites applications for a visiting fac-

University of Hawai`i at Hilo
Department of Computer Science and Engineering

ASSISTANT PROFESSOR OF COMPUTER SCIENCE (3 POSITIONS) to teach introductory and advanced undergraduate courses in computer science, perform scholarly research and professional service in computer science, participate in on-going course and curriculum development activities, advise students, carry out departmental and University service activities. Requires Ph.D. (by May 31, 2009) in Computer Science or closely related discipline from an accredited college or university, evidence of ability to teach computer science courses effectively at the undergraduate level, evidence of commitment to ongoing scholarship in computer science, ability to communicate clearly in written and spoken English. Areas of interest include, but are not limited to, database, theory of computing, security, data modeling and visualization, and high-performance computing. For complete description, qualifications, and application requirements visit www.uhh.hawaii.edu/uhh/hr/jobs.php. University of Hawai`i at Hilo is an EEO/AA Employer D/M/V/W.

ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to *acmmediasales@acm.org*. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Rates: $325.00 for six lines of text, 40 characters per line. $32.50 for each additional line after the first six. The MINIMUM is six lines.

Deadlines: Five weeks prior to the publication date of the issue (which is the first of every month). Latest deadlines: *http://www.acm.org/publications*

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at: *http://campus.acm.org/careercenter*

Ads are listed for a period of 30 days.

For More Information Contact: ACM Media Sales at 212-626-0654 or *acmmediasales@acm.org*

ulty position (pending budget approval) for the 2009-2010 academic year. We are especially interested in candidates who can contribute to our software engineering expertise in one or more of the following areas: software project management, software architecture, software testing and quality assurance, software maintenance and evolution, and software construction. An advanced degree in computer science, software engineering, or a closely related field is normally required of applicants from academia; exceptions will be made for candidates with appropriate industry experience. Applicants with significant software engineering experience are particularly encouraged to apply.

Rose-Hulman is a highly selective, coeducational, primarily undergraduate college of engineering and science, with a vision to be the best institution of its kind in the world. Faculty members are expected to be outstanding teachers and professionally active. The department offers B.S. degrees in software engineering and computer science. These programs are accredited by the Engineering Accreditation Commission and the Computing Accreditation Commission, respectively, of ABET, Inc. Program descriptions may be found in the "Academics" area of http://www.rose-hulman.edu/.

Applicants should submit a cover letter, curriculum vitae, and a statement on their teaching philosophy to: https://jobs.rose-hulman.edu/. Three letters of recommendation should be submitted by the applicant's references through the same website. EEO/AA.

Rutgers University
Tenure Track Faculty Position in Computational Biomedicine, Imaging and Modeling

The Rutgers University Department of Computer Science <http://www.cs.rutgers.edu/> and the Center for Computational Biomedicine, Imaging and Modeling <http://www.cbim.rutgers.edu/> (CBIM) seek applicants in large scale data analytics for a tenure-track assistant professor faculty position starting September 2009. We're particularly interested in synergy with CBIM and thus we're excited about receiving applications in all areas of large scale data analytics, such as computational science, physics-based modeling, visualization and human-computer interaction, computer vision, machine learning and data mining. Applicants whose research does not clearly demonstrate applications in large scale data analytics will not be considered. Rutgers University offers an exciting and multidisciplinary research environment and encourages collaborations between Computer Science and other disciplines.

Applicants for this research/teaching position must, at minimum, be in the process of completing a dissertation in Computer Science or a closely related field, and should show evidence of exceptional research promise, potential for developing an externally funded research program, and commitment to quality advising and teaching at the graduate and undergraduate levels. Hired candidates who have not defended their Ph.D. by September will be hired at the rank of Instructor, and must complete the Ph.D. by December 31, 2009 to be eligible for tenure-track title retroactive to start date. Applicants should send their curriculum vitae, a research statement addressing both past work and future plans, a teaching statement, and arrange for four letters of recommendation to be sent on their behalf to hiring.cbim@cs.rutgers.edu. If electronic submission is not possible, hard copies of the application materials may be sent to:

CS/CBIM Search, c/o Mary Hoffman
Computer Science Department
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854

Applications should be received by February 15, 2009, for full consideration.

Rutgers University is an Affirmative Action/Equal Opportunity Employer. Women and minority candidates are especially encouraged to apply.

Rutgers University
Tenure-Track Position

The Department of Computer Science at Rutgers University invites applications for tenure-track faculty positions at the rank of Assistant, Associate or full Professor, with appointments starting in September 2009, subject to the availability of funds. All areas in experimental computer systems will be considered, but special emphasis will be given to pervasive computing and computer architecture.

Applicants for this research/teaching position must, at minimum, be in the process of completing a dissertation in Computer Science or a closely related field, and should show evidence of exceptional research promise, potential for developing an externally funded research program, and commitment to quality advising and teaching at the graduate and undergraduate levels. Hired candidates who have not defended their Ph.D. by September will be hired at the rank of Instructor, and must complete the Ph.D. by December 31, 2009 to be eligible for tenure-track title retroactive to start date.

Applicants should send their curriculum vitae, a research statement addressing both past work and future plans, and a teaching statement to hiring@cs.rutgers.edu, and have three letters of recommendation sent to this same address. If electronic submission is not possible, hard copies of the application materials may be sent to:

Mary Hoffman
Computer Science Department
Rutgers University
110 Frelinghuysen Road
Piscataway, NJ 08854

Applications should be received by January 20, 2009 for full consideration.

Southern Illinois University Edwardsville
Assistant/Associate Professor

The Department of Computer Science at Southern Illinois University Edwardsville invites applications for one tenure-track position at the Assistant or Associate Professor levels beginning in August 2009. All areas will be considered, however, these areas are of particular interests: Software Engineering, HCI, Database, and Robotics. A doctorate in Computer Science or a related field at the time of appointment is required.

Located twenty miles from downtown St. Louis, SIUE has an enrollment of over 13,000 students. At SIUE, good teaching is of primary importance and research leading to publication is required for promotion and tenure.

Part of the SIUE School of Engineering, the Department offers an ABET-accredited B.S. degree program as well as B.A. and M.S. degree programs. The twelve full-time faculty and nearly 200 majors employ labs for software engineering, HCI, robotics, virtual reality, and networking, all maintained by the department. For more details, visit us at http://www.cs.siue.edu/.

To apply, send a letter of application, résumé, transcripts, and three letters of reference to: Computer Science Search Committee, SIUE Campus Box 1656, Edwardsville, IL 62026-1656. Review of applications will begin on January 26, 2009.

SIUE is an Equal Opportunity/Affirmative Action Employer. Applications from women and minorities are especially encouraged.

Southern Polytechnic State University
Tenure-track positions in Computing

We are seeking to fill one or two tenure-track Assistant Professor positions (other ranks considered). Candidates with interests in multiple areas, including networking, information assurance and security, mobile computing, embedded systems, computer architecture, programming languages, and/or system administration, are sought. Applicants must hold a relevant Ph.D., be committed to excellence in teaching, and have substantial research potential. SPSU is located in the metro Atlanta area, which offers excellent quality of life. For more details, go to http://www.spsu.edu/hr/AssistantProfessor-CSWE.htm.

Texas A&M University
Department of Computer Science and Engineering
Tenure-Track Faculty Positions

Applications* are invited for tenure-track positions, starting fall 2009, in the Department of Computer Science and Engineering of the Dwight Look College of Engineering at Texas A&M University. As part of a long-term plan to increase the size and improve quality, the department is expanding with an assistant professor position in the area of robotics. Top candidates in other areas at all professor levels will also be considered. Candidates must have a Ph.D. degree and will be expected to teach, perform research, and supervise graduate students.

The Department of Computer Science and Engineering has 40 tenured, tenure-track faculty and 4 senior lecturers. Texas A&M University CS faculty members are well recognized for contributions to their fields. The department currently has one National Academy of Engineering member, five IEEE Fellows, one ACM Fellow and over ten PYI/NYI/CAREER awardees. Additional information about the department can be found at www.cs.tamu.edu.

Texas A&M University CS faculty applicants should apply online at

apply.cs.tamu.edu/tenuretrack.

For questions about the positions, contact: search@cs.tamu.edu,

* Applications are welcome from dual career couples.

Texas A&M University is an equal opportunity/affirmative action employer and actively seeks candidacy of women and minorities.

The Ohio State University
Assistant Professor

The Department of Computer Science and Engineering (CSE), The Ohio State University, invites applications for two tenure-track positions at the Assistant Professor level. The positions are open to all CSE areas (artificial intelligence, graphics and animation, networking, software engineering and programming languages, systems, and theory) with priority consideration given to candidates with cross-cutting interests in database systems and machine learning (e.g. data mining) and those in theory (broadly defined).

The department is committed to enhancing faculty diversity; women, minorities, or individuals with disabilities are especially encouraged to apply.

Applicants should hold or be completing a Ph.D. in CSE or a closely related field, and have a commitment to and demonstrated record of excellence in research as well as a commitment to excellence in teaching.

The department maintains and encourages multi-disciplinary research and education activities within and outside The Ohio State University.

To apply, please submit your application via the online database. The link can be found at:

http://www.cse.ohio-state.edu/department/positions.shtml

Review of applications will begin in January and will continue until the positions are filled.

The Ohio State University is an Equal Opportunity/Affirmative Action Employer.

Towson University
Assistant Professor

The Dept. of Computer and Information Sciences at Towson University (Towson, Maryland) invites applications for a new tenure-track assistant professor position to begin Fall 2009. The responsibilities include teaching undergrad/grad courses, conducting research, seeking grants and collaborating with government agencies and the private sector, establishing a BS program in Information Technology (IT), and supervising graduate (MS and/or D.Sc.) student theses and projects. Applicants must have a doctorate in CS, CIS, IT, or a related area. Applicants with industry experience and research focus in the areas of web technology, security, or other AIT areas are strongly encouraged to apply.

More information is available at http://www.towson.edu/cosc/positions.asp and applications or inquires can be sent to coscsearch@towson.edu

Trinity College
Computer Science Department
Visiting Assistant Professor

Applications are invited a 2 year visiting assistant professor position with a possibility of a 1 year renewal beginning fall of 2009. A Ph.D. in Computer Science or related field is preferred, but ABDs will be considered. All research areas will be considered. The teaching load at Trinity is 5 courses per year. Trinity College is an Equal Opportunity/Affirmative Action Employer. Women and minorities are encouraged to apply. Applicants should send a vita and arrange for 3 letters of reference to be sent to: Search Committee, Computer Science Department, Trinity College, Hartford, CT 06106 or to nancy.fleming@trincoll.edu.

Consideration of applications will begin February 1. Inquiries can be addressed to Madalene.Spezialetti@trincoll.edu. Additional information about the department can be found at http://www.trincoll.edu/~cpsc/

University Corporation for Atmospheric Research
Scientist I

The Computational and Information Systems Laboratory (CISL) at the National Center for Atmospheric Research (NCAR) in Boulder, Colorado, seeks an individual to conduct research and development in the fields of computational science, scientific computing, high-performance computing, and computing systems as it relates to NCAR's mission. We are particularly interested in scientists with experience and expertise in the following areas: technologies and techniques for petascale computing, massively parallel algorithms, optimization for multi/many-core systems, accelerator technologies (e.g., GPGPUs and FGPAs), and software engineering for parallel applications.

Initial consideration will be given to applications received prior to 3:00 p.m. MST on Monday, January 5, 2009. Thereafter, applications will be reviewed on an as-needed basis. Apply online at www.ucar.edu (reference job #9028). We value diversity. AA/EOE

Application materials should include: A statement of research interest; a Curriculum Vitae (CV); and a list of referees from whom we may expect letters of recommendation to be sent. Please ask referees to submit their letters to: SCI09T-DD@ucar.edu

View detailed job description at www.ucar.edu at the Jobs & Opportunities/Careers @ UCAR link.

University of California, Irvine
Donald Bren School of Information and Computer Sciences
Faculty Position in Organizational Studies of Information Technology (Tenure Track)

The Department of Informatics at the University of California, Irvine (UCI) is seeking excellent candidates for a tenure-track position in organizational studies of information technology starting in July 2009.

The position is targeted at the rank of assistant professor, but exceptional candidates at all ranks will be considered. Organizational studies of information technology are a particular strength of the Department, and we are looking for candidates who both broaden and deepen our vision. More information on this and other positions is available at http://www.ics.uci.edu/employment/employ_faculty.php.

University of California, Santa Barbara
Veeco Endowed Chair in Scanning Probe Techniques

UC Santa Barbara is now accepting applications and nominations for the inaugural holder of the Veeco Endowed Chair in Engineering and the Sciences. Candidates for the endowed chair must hold a Ph.D. in a field that has benefited from nanoscale characterization, including but not limited to Materials Science, Engineering, Physics, Chemistry, Biology, Energy, Multidisciplinary Nanoscience or related discipline. Candidates with a research emphasis on scanning probe technologies, and who have developed new scanning probe methodologies to advance the state of the art in their field, are strongly encouraged to apply. The holder of the Veeco Chair may be appointed at a senior, tenured level, with an anticipated start date of July 1, 2009, either in the College of Engineering or in the Division of Mathematical, Life, and Physical Sciences. For more details, please see http://www.engineering.ucsb.edu/positions.

All applicants and nominees will be held in confidence by the Search Committee. Please send CV, a statement of research and teaching interests, and a representative recently published paper to Dean Matthew Tirrell, attn: Veeco Endowed Chair Search, College of Engineering, Universityof California, Santa Barbara, CA 93106, or to VeecoChair@engineering.ucsb.edu. Applications will be shared with appropriate campus departments in the sciences and engineering fields. Apply by March 31, 2009 for primary consideration; however, position will remain open until filled.

An EO/AA Employer.

University of California, San Diego
Tenured or Tenure-track Faculty Positions

The UCSD Department of Computer Science and Engineering (CSE) seeks to fill multiple faculty positions at tenured or tenure-track positions. We invite applications at all levels in all areas of computer science and computer engineering, with particular interest in the areas of algorithms and theory, bioinformatics, and graphics. Faculty positions in other than the tenured/tenure-track series are also available. Exceptional candidates in all areas will be given serious consideration. The department is looking for applicants with outstanding research credentials. Successful applicants are expected to lead a vigorous research program and to have a strong commitment to teaching. A Ph.D. in computer science or a related area is desired. Salary and rank will be commensurate with qualifications in conformance with University of California policies. CSE is home to over 50 faculty and 300 graduate students who together span a range of research areas in computer science, computer engineering, and bioinformatics. In addition, the department works closely with the Center for Networked Systems (CNS), the California Institute for Telecommunications and Information Technology (Calit2), the San Diego Supercomputer Center (SDSC), and the Center for Wireless Communications (CWC), which provides unique opportunities and resources to our members. Finally, the department has recently occupied a newly constructed facility that provides significant space resources for a wide range of research activities. More information about the department can be found on the Web at http://www.cs.ucsd.edu/

Application Procedure
For details on how to apply click here
http://www.cse.ucsd.edu/home/prospective faculty/currentad.html

Closing Date: We encourage candidates to apply online by clicking on the URL listed above. Review of applications will begin on January 7, 2009 and will continue until positions have been filled.

Applicants are asked to include a personal statement summarizing teaching experience. Please highlight any leadership efforts, and/or contributions to promotion of diversity in the workplace.

UCSD is an Equal Opportunity/Affirmative Action Employer with a strong institutional commitment to the achievement of excellence and diversity among its faculty and staff. Women and minority applicants, veterans and persons with disabilities are encouraged to apply.

For applicants with interest in spousal/partner employment, please see our Website for the UCSD Partner Opportunities Program.

University of Central Oklahoma
Tenure Track Assistant Professor of John T. Beresford Endowed Chair

The University of Central Oklahoma, College of Mathematics and Science, Department of Computer Science is seeking a full-time tenure track assistant professor for the John Taylor Beresford endowed chair. This endowment includes funds to be used for research. The requirements of the position include teaching undergraduate computer science courses, performing research with assistance from undergraduate students, and serving on department/college/university committees. The successful candidate should have an earned Ph.D. in Computer Science as of August 1, 2009. Preference will be given to applications with a background which includes computer graphics, software engineering, or computer gaming. The successful candidate should have experience in teaching and in research.

Applications are being accepted online https://jobs.uco.edu. Refer to position #A000443. Open until filled. Preference given to applications received before February 16, 2009.

University of Colorado at Boulder
Faculty Positions in Computational Biology

The University of Colorado at Boulder invites applications for two tenure-track faculty positions in the broad areas of computational biology and bioinformatics, under the auspices of the Colorado Initiative in Molecular Biotechnology (cimb.colorado.edu).

Individuals with interests in developing and applying computational or mathematical methods to biological systems are encouraged to apply.

CIMB is a program that integrates faculty from the departments of Applied Mathematics; Chemical & Biological Engineering; Chemistry & Biochemistry; Computer Science; Ecology and Evolutionary Biology; Integrative Physiology; Mechanical Engineering; Molecular, Cellular & Developmental Biology; and Physics. A successful candidate may be rostered in any one of these departments. The positions are at the Assistant Professor level, although outstanding senior candidates at higher ranks may be considered. Candidates must have a Ph.D. degree and a demonstrated commitment to teaching at undergraduate and graduate levels, and will be expected to develop an internationally recognized research program.

Applicants should submit a curriculum vitae, statements of research and teaching interests, and arrange to have three letters of reference electronically to: https://www.jobsatcu.com, posting #805756. Review of applications will begin on or about December 1, 2008 and will continue until the position is filled. For more information or assistance in submission of materials, contact Janice McClintock (Janice.Mcclintock@colorado.edu). The University of Colorado is sensitive to the needs of dual career couples, and is committed to diversity and equality in education and employment. See www.Colorado.edu/ArtsSciences/Jobs/ for full job description.

University of Denver
Professor and Department Chair

The Department of Computer Science (CS) at the University of Denver (DU) is seeking a dynamic and visionary individual from business or academia to lead the department during this expansion phase of the School of Engineering and Computer Science. Through its strategic planning, the faculty of our CS department have identified Software Engineering, Game Development, and Cyber Security as the key focus areas for the department. Our CS department benefits from a top quality faculty, strong partnership with industry, strong collaborations with other colleges within DU and internationally. The CS department offers degrees in both traditional and contemporary areas such as undergraduate degree in gaming, and graduate degree in Computer Science Systems Engineering. The primary focus of this new department chair will be on both educational and research programs at graduate and undergraduate levels. DU is a private university with a strong history of academic excellence, small classes, and emphasis on student engagement at all levels. DU is the oldest university in Colorado and its campus is located in the Denver metro area.

Individuals with a strong record of research, scholarship and excellence in teaching are encouraged to apply by sending their resume, statement of interest, and a list of five references to www.dujobs.org. PhD or PhD candidate in computer science or related areas and some level of leadership experience are required. The University of Denver is an AA/EOE.

University of North Carolina at Charlotte
Department of Software and Information Systems
DICyDER Center Director

The Department of Software and Information Systems at UNC Charlotte seeks to hire a tenure-track faculty member at the associate level to serve as Director of the recently established

Center for Digital Identity and Cyber Defense Research (DICyDER), http://www.dicyder.uncc.edu/. DICyDER's mission is to add value to the university, community, and society through innovative educational programs, research and development in the areas of information integration, security, and privacy. The Director will be responsible for leading a strong research program by communicating research vision, planning and implementing research strategy, facilitating contract acquisition and relationship development, and providing project and group management.

The Department of Software and Information Systems is dedicated to research and education in Software Engineering and Information Technology applications, with emphasis in the areas of Information Integration & Environments and Information Security & Assurance; it offers degrees at the Bachelors, Masters, and Ph.D. levels. Current faculty members have strong research programs with substantial funding from both federal agencies and industrial partners.

Salary will be highly competitive. Applicants must have a Ph.D. in Computer Science, Information Technology, Software Engineering, or a related field, as well as a strong commitment to research and education. For further details please visit http://www.sis.uncc.edu/. Application review will start in January 2009.

Applications must be submitted online at https://jobs.uncc.edu/. To the application, attach a cover letter, curriculum vitae, a statement of teaching interests, a statement of research interests, copies of three representative scholarly publications, and a list of four references. For questions or additional information, email search-sis@uncc.edu.

Women, minorities and individuals with disabilities are encouraged to apply. UNC Charlotte is an Equal Opportunity/Affirmative Action employer.

University of North Carolina at Charlotte
Department of Software and Information Systems
Tenure-Track Faculty Positions

The Department of Software and Information Systems at UNC Charlotte invites applicants for multiple tenure-track faculty positions at both the assistant and associate levels. The Department is dedicated to research and education in Software Engineering and Information Technology applications, with emphasis in the areas of Information Integration & Environments and Information Security & Assurance; it offers degrees at the Bachelors, Masters, and Ph.D. levels. Current faculty members have strong research programs with substantial funding from both federal agencies and industrial partners. The department is particularly interested in faculty with research expertise in: Trusted Software Development, Software Engineering, or Modeling & Simulation. Highly qualified candidates in other areas will also be considered.

Salary will be highly competitive. Applicants must have a Ph.D. in Computer Science, Information Technology, Software Engineering, or a related field, as well as a strong commitment to research and education. For further details please visit http://www.sis.uncc.edu/. Application review will start in January 2009.

Applications must be submitted online at https://jobs.uncc.edu/. To the application, please attach a cover letter, curriculum vitae, a statement of teaching interests, a statement of research interests, copies of three representative scholarly publications, and a list of four references. For questions or additional information, please email search-sis@uncc.edu.

University of Rochester
Tenure Track Faculty Positions

The Department of Computer Science at the University of Rochester invites applications for tenure track faculty positions. We seek PhD level candidates in networking, HCI, graphics, and/or machine learning. In addition, we invite applications for a joint Computer Science/Electrical and Computer Engineering position in computer systems and circuits. For full job descriptions and application procedures, see http://www.cs.rochester.edu/recruit.

University of Texas at Dallas
Computer Science Department Head

The Erik Jonsson School of Engineering and Computer Science at the University of Texas at Dallas (UTD) invite nominations and applications for the position of Head of the Department of Computer Science. Candidates for the position must have a Ph.D. degree in Computer Science or a related field.

Qualified candidates must have a demonstrated dedication to undergraduate and graduate education, a strong record of scholarly and professional achievements, leadership and organizational skills and overall qualifications commensurate with the rank of full professor in the Department. The selected candidate will be responsible for recruiting new faculty in the Department, curriculum development and strengthening the educational and research programs in both traditional and interdisciplinary areas, as well as areas capitalizing on existing strengths and excellence of the Jonsson School of Engineering.

One of the largest departments of its kind in the country, UTD's Department of Computer Science features an internationally recognized faculty, nearly 1,500 students and a 150,000-square-foot building with modern classrooms and state-of-the-art laboratories. Areas of research include cybersecurity, networking, embedded software, programming languages and systems, human language technology, software engineering, intelligent systems and theory. CS faculty and students publish more than 350 research papers annually. The CS Department includes multiple NSF Career Award winners, and is one of only a few departments in the U.S. offering B.S., M.S. and Ph.D. degrees in software engineering.

The University of Texas at Dallas is situated in Richardson, one of the most attractive suburbs of the Dallas metropolitan area with several hundred high-tech companies within a few miles of the campus, including Texas Instruments, Lockheed Martin, Raytheon, Nortel Networks, Alcatel, Ericsson, Hewlett-Packard, Samsung, Fujitsu, Cisco Systems, EDS, Zyvex, and Intervoice. Opportunities for joint university-industry research projects are excellent. The Erik Jonsson School is experiencing a very rapid growth as part of a $300 million program of funding from public and private sources. As a result, the school is expanding its existing programs, recruiting outstanding faculty and Ph.D. students, increasing funded research, and establishing new programs. A $100 million state-of-the art building for interdisciplinary research in science and engineering was recently inaugurated.

For more information about the Jonsson School of Engineering visit http://www.ecs.utdallas.edu or send e-mail to Dr. Bhavani Thuraisingham, Chair (bxt043000@utdallas.edu). The search committee will begin evaluating applications as soon as they are received and will continue until the position is filled.

Applicants should mail their resume with a list of at least three academic or professional references to:

Academic Search # 7093
The University of Texas at Dallas
800 W. Campbell Rd., AD 42
Richardson, TX 75080-3021

Indication of gender and ethnic origin for affirmative action statistical purposes is requested as part of the application process but is not required for consideration. The University of Texas at Dallas is an Equal Opportunity Affirmative Action employer and strongly encourages applications from candidates who would enhance the diversity of the University's faculty and administration.

University of Toronto
Assistant Professor (tenure track), Computer Science

The Department of Computer Science, University of Toronto, invites applications for a tenure-track appointment at the rank of Assistant Professor, to begin July 1, 2009.

We are seeking applicants whose interdisciplinary research expands traditional boundaries of computer science. Exceptional applicants in all areas of computer science meeting this criterion will be considered. Particular emphasis will be placed on such applicants in human-computer interaction (HCI) and closely-related areas.

The University of Toronto is an international leader in computer science research and education, and the department enjoys strong interdisciplinary ties to other units within the University.

Candidates should have (or be about to receive) a Ph.D. in computer science or a related field. They must demonstrate an ability to pursue innovative research at the highest level, and a strong commitment to teaching.

Salaries are competitive with our North American peers and will be determined according to the successful applicant's experience and qualifications. Toronto is a vibrant and cosmopolitan city, one of the most desirable in the world in which to work and live. It is also a major centre for advanced computer technologies; the department has strong interaction with the computer industry.

To apply for this position, please visit http://recruit.cs.toronto.edu/ and follow the instructions.

The review of applications will commence on December 15, 2008. To ensure full consideration, applications should be received by January 15, 2009.

The University of Toronto is strongly committed to diversity within its community and especially welcomes applications from visible minority group members, women, Aboriginal persons, persons with disabilities, members of sexual minority groups, and others who may contribute to the further diversification of ideas.

All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

University of Washington
Computer Science & Engineering and Electrical Engineering
Tenure-Track and Research Faculty

The University of Washington's Department of Computer Science & Engineering and Department of Electrical Engineering have jointly formed a new UW Experimental Computer Engineering Lab (ExCEL). In support of this effort, the College of Engineering has committed to hiring several new faculty over the forthcoming years. All positions will be dual appointments in both departments (with precise percentages as appropriate for the candidate). This year, we have two open positions, and encourage exceptional candidates in computer engineering, at tenure-track Assistant Professor, Associate Professor, or Professor, or Research Assistant Professor, Research Associate Professor, or Research Professor to apply. A moderate teaching and service load allows time for quality research and close involvement with students. The CSE and EE departments are co-located on campus, enabling cross department collaborations and initiatives. The Seattle area is particularly attractive given the presence of significant industrial research laboratories, a vibrant technology-driven entrepreneurial community, and spectacular natural beauty. Information about ExCEL can be found at <http://www.excel.washington.edu>.

We welcome applications in all computer engineering areas including but not exclusively: atomics scale devices & nanotechnology, implantable and biologically-interfaced devices, synthetic molecular engineering, VLSI, embedded systems, sensor systems, parallel computing, network systems, and technology for the developing world. We expect candidates to have a strong commitment both to research and teaching. ExCEL is seeking individuals at all career levels, with appointments commensurate with the candidate's qualifications and experience. Applicants for both tenure-track and research positions must have earned a PhD by the date of appointment.

Please apply online at <http://www.excel.washington.edu/jobs.html> with a letter of application, a complete curriculum vitae, statement of research and teaching interests, and the names of at least four references. Applications received by January 31st, 2009 will be given priority consideration. Open positions are contingent on funding.

The University of Washington was awarded an Alfred P. Sloan Award for Faculty Career Flexibility in 2006. In addition, the University of Washington is a recipient of a National Science Foundation ADVANCE Institutional Transformation Award to increase the participation of woman in academic science and engineering careers. We are building a culturally diverse faculty and encourage applications from women and minority candidates. The University of Washington is an affirmative action, equal opportunity employer.

University of Washington
Computer Science & Engineering
Tenure-Track, Research, and Teaching Faculty

The University of Washington's Department of Computer Science & Engineering has one or more open positions in a wide variety of technical areas in both Computer Science and Computer Engineering, and at all professional levels. A moderate teaching load allows time for quality research and close involvement with students. Our space in the Paul G. Allen Center for Computer Science & Engineering provides opportunities for new projects and initiatives. The Seattle area is particularly attractive given the presence of significant industrial research laboratories as well as a vibrant technology- driven entrepreneurial community that further enhances the intellectual atmosphere. Information about the department can be found on the web at http://www.cs.washington.edu.

We welcome applicants in all research areas in Computer Science and Computer Engineering including both core and inter-disciplinary areas. Areas of interest include (but are not limited to) security, machine learning, theory, and systems. We expect candidates to have a strong commitment both to research and to teaching. The department is primarily seeking individuals at the tenure-track Assistant Professor rank; however, under unusual circumstances and commensurate with the qualifications of the individual, appointments may be made at the rank of Associate Professor or Professor. We may also be seeking non-tenured research faculty at Assistant, Associate and Professor levels, postdoctoral researchers (Research Associates) and part-time and full-time annual lecturers and Sr. Lecturers. Applicants for both tenure-track and research positions must have earned a doctorate by the date of appointment; those applying for lecturer positions must have earned at least a Master's degree. Research Associates, Lecturers and Sr. Lecturers will be hired on an annual or multi-annual appointment. All University of Washington faculty engage in teaching, research and service.

Please apply online at <http://www.cs.washington.edu/news/jobs.html>with a letter of application, a complete curriculum vitae, statement of research and teaching interests, and the names of four references. Applications received by February 1, 2009 will be given priority consideration. Open positions are contingent on funding.

The University of Washington was awarded an Alfred P. Sloan Award for Faculty Career Flexibility in 2006. In addition, the University of Washington is a recipient of a National Science Foundation ADVANCE Institutional Transformation Award to increase the participation of women in academic science and engineering careers. We are building a culturally diverse faculty and encourage applications from women and minority candidates. The University of Washington is an affirmative action, equal opportunity employer.

University of Waterloo
David R. Cheriton School of Computer Science
Faculty Positions in Information Systems

The University of Waterloo invites applications for one or two tenure-track or tenured faculty positions in the David R. Cheriton School of Computer Science, in the areas of information systems. Candidates at all levels of experience are encouraged to apply. Preference will be given to those who focus on health informatics as an application area. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, contribute to a newly-created Master's program in health informatics, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute to the overall development of the School. A Ph.D. in Computer Science, or equivalent, is required, with evidence of excellence in teaching and research. Rank and salary will be commensurate with experience, and appointments are expected to commence during the 2009 calendar year.

With over 70 faculty members, the University of Waterloo's David R. Cheriton School of Computer Science is the largest in Canada. It enjoys an excellent reputation in pure and applied research and houses a diverse research program of international stature. Because of its recognized capabilities, the School attracts exceptionally well-qualified students at both undergraduate and graduate levels. In addition, the University has an enlightened intellectual property policy which vests rights in the inventor: this policy has encouraged the creation of many spin-off companies including iAnywhere Solutions Inc., Maplesoft Inc., Open Text Corp and Research in Motion. Please see our website for more information: http://www.cs.uwaterloo.ca.

Applications should be sent by electronic mail to cs-recruiting@cs.uwaterloo.ca, or by post to

Chair, Advisory Committee on Appointments
David R. Cheriton School of Computer Science
200 University Avenue West
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

An application should include a curriculum vitae, statements on teaching and research, and the names and contact information for at least three referees. Applicants should ask their referees to forward letters of reference to the address above. Applications will be considered as soon as possible after they are complete, and as long as positions are available.

The University of Waterloo encourages applications from all qualified individuals, including women, members of visible minorities, native peoples, and persons with disabilities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

University of Waterloo
David R. Cheriton School of Computer Science
Faculty Position in Software Engineering

The University of Waterloo invites applications for a tenure-track or tenured faculty position in the David R. Cheriton School of Computer Science, in the area of software engineering. Candidates at all levels of experience are encouraged to apply. Successful applicants who join the University of Waterloo are expected to develop and maintain a productive program of research, attract and develop highly qualified graduate students, provide a stimulating learning environment for undergraduate and graduate students, and contribute

LAST BYTE

[CONTINUED FROM P. 112] of information flow. What can or can we not guarantee with respect to preserving the privacy of citizens? How can we control what happens to the information we divulge about ourselves? What would be the equivalent of an impossibility result in privacy, in the way that we have for reliability?

Do you have time these days to actually do research of your own?

Yes, I do. Every program and division director has that opportunity, and I think it's really important, because it keeps you current, it keeps you abreast of what's happening, and it keeps you engaged. I try to devote the equivalent of at least one day a week to research—talking and emailing with my students, and working at nights and weekends. I also make it a point to try to go back to Carnegie Mellon at least once or twice a month.

Let's talk about computational thinking, the term you coined to describe computer scientists' unique way of approaching, understanding, and solving problems.

Computational thinking encompasses all the benefits of computational models and methods and tools and the abstractions that we use. It's not just our metal tools, it's our mental tools—that's a phrase I like to use.

What sort of reception has it gotten?

There's been an incredible groundswell of support and enthusiasm from around the world, from all sorts of cultures and institutions. And there's been a lot of interest recently in computational thinking with respect to education. Many colleges and universities are revisiting their undergraduate curricula and realizing that an introductory computer science course, an introduction to Java programming, is hardly the best we can offer.

I understand you're also focusing on the K–12 level.

Absolutely. The Computer Science and Telecommunications Board, through a partnership with NSF, is going to be planning a series of workshops to address exactly that question: what are effective ways of learning, or teaching, computational thinking by, or to, children.

Leah Hoffmann is a Brooklyn, NY-based science and technology writer.

DOI:10.1145/1435417.1435440

Leah Hoffmann

Q&A
The Upper Limit

Jeannette Wing talks about clusters, creativity, and the power of computational thinking.

A PROFESSOR OF computer science at Carnegie Mellon University, Jeannette Wing has long been interested in the power of abstraction. She is a leading researcher in the area of formal methods, and has worked on diverse problems in software specification and verification, security, concurrent and distributed systems, and programming languages. Wing is also deeply engaged with a concept she calls computational thinking: the way computer scientists use decomposition, recursion, and algorithms to tackle difficult problems. As the current assistant director for computer and information science and engineering at the National Science Foundation (NSF), she now has the opportunity to put her talent for abstraction to use and help shape the future of the field.

You're about a year and a half into your tenure at NSF. What's it been like?

It's been an eye-opening experience. It gives me a much better appreciation of the university/industry/government ecosystem, and how we all need to work together to advance the frontiers of science and engineering.

What role do you play within the organization?

I wear two hats. One is to act as a spokesperson for the field, a scientist who represents the whole computing community—and let me just use the word "computing" to stand for all of computer and information science and engineering. At the same time, I

PHOTOGRAPH BY RICHARD KELLY PHOTOGRAPHY

also have the opportunity to help guide and shape the frontiers of our field.

How's that?

NSF is really a bottom-up organization. We listen to the computing community to find out what the trends are, where they're going, and what their research interests are, and we try to respond to their needs.

"Computational thinking encompasses all the benefits of computational models and methods and tools and the abstractions we use."

Let's talk about some of your most recent initiatives, such as the Cluster Exploratory program.

The Cluster Exploratory program enables us to offer software and services running on a Google-IBM cluster to the entire computing community. For a long time, we could see that there was something very exciting happening in industry with respect to these large data centers. But no university can really afford to build its own center and run it and pay for the power. Now, finally, researchers can access these resources and run their experiments at scale.

There's also a second cluster at the University of Illinois at Urbana-Champaign (UIUC).

Yes, thanks to HP, Intel, and Yahoo!. It's hosted at UIUC, but it's being made available to all of academia. And this time, it's not just the software and services that we're providing, it's access to the bare machine, so researchers can run experiments that go down to the processing level and below.

Has your involvement with NSF altered your own research interests?

Being here has increased my awareness of the need to do more fundamental research into privacy, and to understand things like information flow and use [CONTINUED ON P. 111]