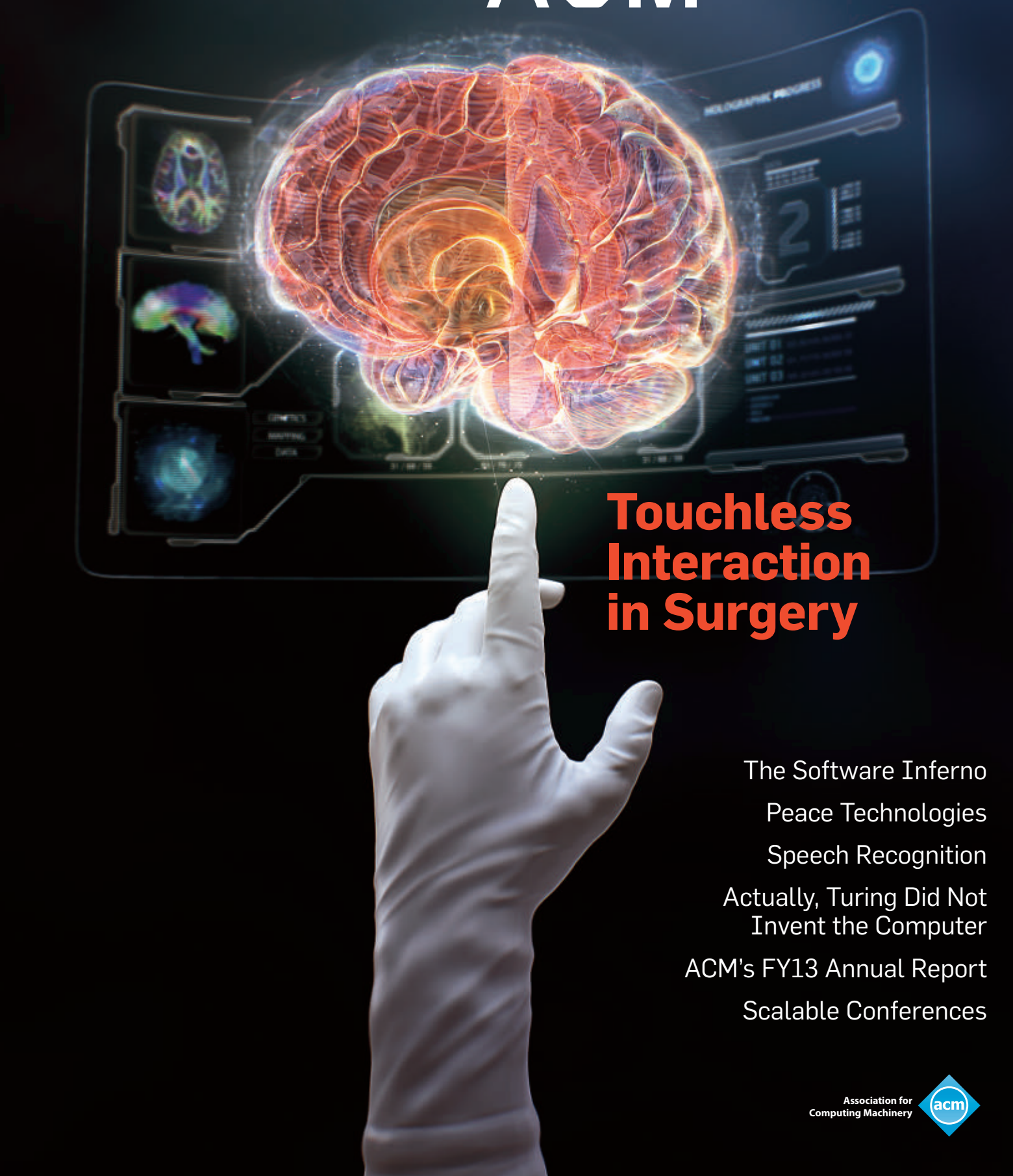
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

01/2014 VOL.57 NO.01



Touchless Interaction in Surgery

The Software Inferno

Peace Technologies

Speech Recognition

Actually, Turing Did Not Invent the Computer

ACM's FY13 Annual Report

Scalable Conferences

Association for Computing Machinery

ICMI 2014

The 16th International Conference on Multimodal Interaction

November 12-16th, 2014

Bogazici University, Istanbul, Turkey

- Multimodal Interaction Processing
- Interactive Systems and Applications
- Modelling Human Communication Patterns
- Data, Evaluation and Standards for Multimodal Interactive Systems
- Urban Interactions

http://icmi.acm.org/2014

Organising Committee

General Chairs
Albert Ali Salah *(Boğaziçi University, Turkey)*
Jeffrey Cohn *(University of Pittsburgh, USA)*
Björn Schuller *(TUM / Imperial College London, UK)*

Program Chairs
Oya Aran *(Idiap Research Institute, Switzerland)*
Louis-Philippe Morency *(University of Southern California, USA)*

Workshop Chairs
Alexandros Potamianos *(University of Crete, Greece)*
Carlos Busso *(University of Texas at Dallas, USA)*

Demo Chairs
Kazuhiro Otsuka *(NTT Comm. Science Lab.s, Japan)*
Lale Akarun *(Boğaziçi University, Turkey)*

Multimodal Grand Challenge Chairs
Dirk Heylen *(University of Twente, The Netherlands)*
Hatice Gunes *(Queen Mary University of London, UK)*

Doctoral Consortium Chairs
Justine Cassell *(Carnegie Mellon University, USA)*
Marco Cristani *(University of Verona, Italy)*

Publication Chairs
Alessandro Vinciarelli *(University of Glasgow, UK)*
Zakia Hammal *(Carnegie Mellon University, USA)*

Publicity Chair
Nicu Sebe *(University of Trento, Italy)*

Sponsorship Chair
Aytül Erçil *(Sabancı University, Turkey)*

Local Organization Chair
Hazım Ekenel *(Istanbul Technical University, Turkey)*

Important Dates

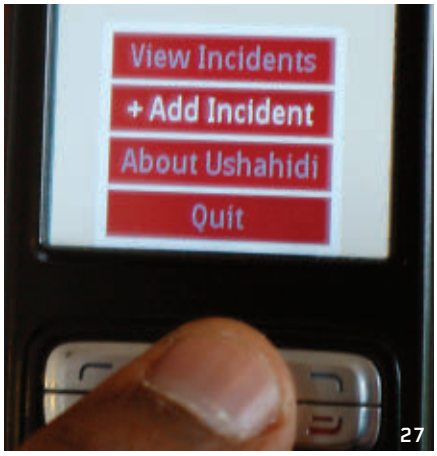
Grand challenge proposals	*January 15th, 2014*
Special session proposals	*March 22nd, 2014*
Workshop proposals	*March 15th, 2014*
Long and short paper submissions	*May 9th, 2014*
Doctoral consortium submissions	*July 1st, 2014*
Demo proposals	*July 15th, 2014*

COMMUNICATIONS OF THE ACM

Departments

Last Byte

News

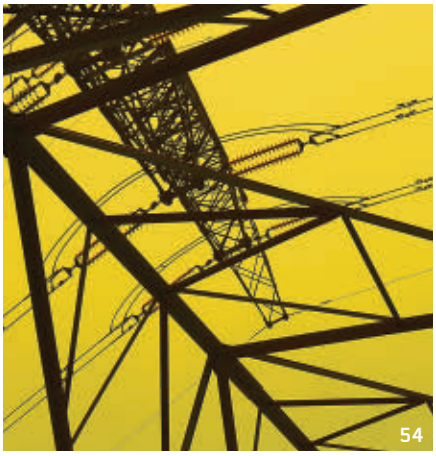


Viewpoints

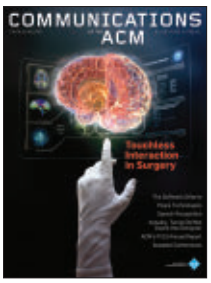
PHOTOGRAPH BY ERIK HERSMAN

Association for Computing Machinery
Advancing Computing as a Science & Profession

Practice



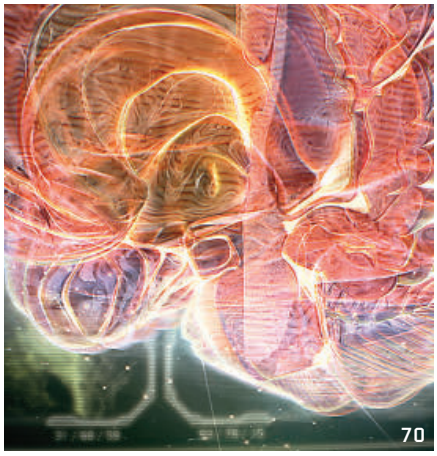
Articles' development led by acmqueue
queue.acm.org



About the Cover: While medical imaging technologies abound, the ability for doctors to interact with visual displays during surgery without compromising the sterility of the operating room has been restrictive. This month's cover story (p. 70) explores the latest technologies that allow surgeons to control and manipulate medical images without touching them. Cover illustration by Kollected.

LEFT, RIGHT: IMAGES FROM SHUTTERSTOCK.COM CENTER: ILLUSTRATION BY KOLLECTED

Contributed Articles



Review Articles

Research Highlights

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

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.

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $100.

ACM Media Advertising Policy
Communications of the ACM and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current Advertising Rates can be found by visiting http://www.acm-media.org or by contacting ACM Media Sales at (212) 626-0686.

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

Printed in the U.S.A.

Association for Computing Machinery

DOI:10.1145/2544169

Moshe Y. Vardi

Scalable Conferences

Adapting computing-research conferences to the growth of the field.

In a recent social-media posting I quoted a blog entry by Michael Mitzenmacher, titled "Easy Now," which opened with the sentence "In the past year or so, I've gotten reviews back on multiple papers with the complaint that the result was too simple." He went on to assert: "From my standpoint, easy is a plus, not a minus." Both the original blog entry and my own posting were heavily commented on, with the general sentiment strongly sympathizing with Mitzenmacher. This unhappiness with the current state of computing-research conferences seems to reflect the general mood in the community, as has been discussed on these pages over the past few years.

A three-day Perspective Workshop on the subject of "Publication Culture in Computing Research" was held at Schloss Dagstuhl in November 2012 (for details, see http://bit.ly/1c9jxAS). A key motivation for the workshop was the observation that in spite of the pervasive dissatisfaction with the status quo, "the community seems no closer to an agreement whether a change has to take place and how to effect such a change." I would have liked to report that we reached agreement that change must take place and we figured out how to effect such a change. Unfortunately, we did not. We did, however, reach agreement on many issues.

One of the main insights developed at the workshop was the computing-research publishing ecosystem—both conferences and journals—has simply failed to scale up with the growth of the field. Consider the following numbers. Between 2002 and 2012, Ph.D. production in computer science and engineering in North America doubled, roughly from 800 to 1,600 (numbers for other parts of the world are not available, regrettably). The number of conference papers published by ACM also roughly doubled, from 6,000 to 12,000. How did we respond to this growth in research production? Simple; instead of doubling the *size* of our conferences we doubled the *number* of conferences. The number of ACM conferences during this period grew from about 80 to almost 160!

We are all aware of the adverse effects of "conference inflation." Instead of serving as community-building events, many conferences have become paper-publishing events, the infamous "journals that meet in hotels." Matching papers and conferences has become more difficult, as reviewers struggle to find reasons to reject papers, such as "the result is too simple." Papers bounce from conference to conference, creating an ever-increasing review workload. It is not uncommon to hear of a paper being rejected summarily from one conference only to receive a best-paper award from another conference.

I find this failure to scale extremely ironic considering how much our discipline is about scaling: higher complexity, larger volumes of data, and larger problems. We have built the Internet, which is about to go interplanetary, but we have failed to scale our own institutions. Considered from that perspective, one path forward in the publication-culture debate is to note the growth of the field and resolve to grow our conferences rather than to continue proliferating them. Imagine SIGPLAN, for example, having, say, two large biannual meetings, rather than the 14 conferences SIGPLAN sponsors now.

A bold proposal along these lines is expressed in the Viewpoint "Publish Now, Judge Later" by Doug Terry on page 44 of this issue. Terry starts with the observation that computing-research conferences today face a reviewing crisis with too many submissions and not enough time for reviewers to carefully evaluate each one. The result is the process, meant to identify the papers of the "highest quality," is itself of questionable quality. In fact, there is evidence that while reviewers may reach consensus on the small fractions of the strongest submissions and the weakest submissions, there is no consensus on the main bulk of the submissions, and the final accept/reject decisions are essentially random.

Terry, therefore, proposes an approach where conferences accept any paper that extends the current body of knowledge, as it is extremely difficult to judge the true significance of any new research result. In this approach, a conference publication is not the *final* publication of a research result, but its *first* publication. Through discussions and follow-on journal publication, the community will eventually reach judgment on the significance of the result.

The change from "reject as default" to "accept as default" would be a significant change to our publication culture. I do not expect to see such a change be adopted quickly or widely. It would be nice, however, to see one computing-research subcommunity be brave enough to experiment with it. To quote a Chinese proverb, "A journey of a thousand miles begins with a single step."

Follow me on Facebook, Google+, and Twitter.

Moshe Y. Vardi, EDITOR-IN-CHIEF

DOI:10.1145/2544172

Vinton G. Cerf

Virtual Reality Redux

I have just returned from a trip to Warsaw where I had the opportunity to visit the Copernicus Science Museum. It was filled with young people racing from one interactive

exhibit to another. For those who may be familiar with the Exploratorium in San Francisco (recently relocated to the Embarcadero), think of the Copernicus Museum as the Exploratorium on steroids. The facility houses an amazing array of interactive exhibits ranging from humaniform robots to an olfactory laboratory that presented hours if not days of opportunity for visitors to explore the real world through carefully crafted displays. Among the most unusual was a laboratory that provided a collection of distilled essences that, when combined, produced pleasant to pungent to downright smelly effects. It included essences from plants and flowers to the anal glands of animals like beavers and civets. (I wondered how these particular essences might have been discovered…).

What does all this have to do with virtual reality?

As I encountered these fascinating experiences drawn from the physical world, I thought about what we have been able to achieve in the virtual world of computing. We create our own realities in this space. We can explore in a simulated way universes that bear little relation to our real world. We can change fundamental physical constants to observe the effects. We can design systems that could work in environments that might exist only in our imagination or might exist only in the hearts of stars. The computable universe is in some sense even larger and diverse than the real one, unless, perhaps, you subscribe to the infinite universe theory.

Just as neural structures in the brain deal only with electro-chemical actions, computers deal with binary bits. The neurons of the brain do not distinguish between input signals coming from ears, nose, eyes, tongue, or fingers. All the senses end up being represented with the same kinds of electro-chemical signals. Unsurprisingly, it is now thought that the same neural structures are used to detect and analyze sensory patterns, regardless of their origin. Computers end up processing binary encoded signals (possibly passing through analog/digital converters) and in both directions. That is, computers receive and interpret incoming digital signals and generate outgoing digital signals, regardless of the ultimate way in which these signals are rendered.

Whether we are typing on a keyboard, fingering a touch pad, or speaking, these media become digital signals suitable for processing. In the other direction, digital signals may be transduced to drive a variety of output media. The modes through which we interact with computers have been evolving toward ever-richer alternatives. The remarkable Microsoft Kinect device is an example in which gestures of all kinds become a new vocabulary through which to communicate with computers. By the same token, output media are growing richer. The worlds of imagination will be rendered by increasingly diverse means, including, one supposes, three-dimensional display technology and so-called "3-D printers." In fact, there is no limit to the potential variety of output media one could imagine. Bone conduction devices cause sound to "materialize" inside the cochlea—as does the speech processor used with cochlear implants. One can begin to imagine other mechanisms that go well beyond today's Google Glass toward direct neuro-electric stimulation of the retina or the optical nerve.

Speculations like this lead one to imagine that Asimov's "visi-sonor" may not be as far-fetched as it seemed when he wrote the Mule in his famous Foundation Trilogy. It also makes one wonder about the potential inherent in today's CAVE display rooms. Perhaps the Star Trek holodeck is not as far in the future as it might seem at first glance. There are already many applications that can process the massive amounts of data taken by magnetic resonance imaging and present the results in three-dimensional format. At the same time, one can readily imagine registering and calibrating analyzed and simulated results overlaid on real images—the classic definition of augmented reality that is already demonstrable. Applications that overlay languages and currency translations over images of menus in the restaurant already exist. Barcode scanners overlay database information on the image of jars of food.

It seems irresistible to predict and inescapable to imagine that in the future we will see broader and more diverse ways in which to render computer output or to capture computer input. Perhaps a 21st-century Descartes will be heard to say, "I think, therefore it is!"

Vinton G. Cerf, ACM PRESIDENT

DOI:10.1145/2541883.2541888

Nominees for Elections and Report of the ACM Nominating Committee

In accordance with the Constitution and Bylaws of the ACM, the Nominating Committee hereby submits the following slate of nominees for ACM's officers. In addition to the officers of the ACM, two Members at Large will be elected. The names of the candidates for each office are presented in random order below:

President (7/1/14–6/30/16):
Alexander L. Wolf, Imperial College London
Ronald Perrott, University of Oxford and Queen's University Belfast

Vice President (7/1/14–6/30/16):
Vicki L. Hanson, Rochester Institute of Technology and University of Dundee
Eugene H. Spafford, Purdue University

Secretary/Treasurer (7/1/14–6/30/16)
David A. Wood, University of Wisconsin
Erik Altman, IBM TJ Watson Research Center

Members at Large (7/1/14–6/30/18):
Victor Bahl, Microsoft Research, Redmond
Per Stenström, Chalmers University of Technology, Sweden
Cherri Pancake, Oregon State University
Madhavan Mukund, Chennai Mathematical Institute, India

The Constitution and Bylaws provide that candidates for elected offices of the ACM may also be nominated by petition of one percent of the Members who as of **November 1** are eligible to vote for the nominee. Such petitions must be accompanied by a written declaration that the nominee is willing to stand for election. The number of Member signatures required for the offices of President, Vice President, Secretary/Treasurer and Members at Large, is 753.

The Bylaws provide that such petitions must reach the Elections Committee before **January 31**. Original petitions for ACM offices are to be submitted to the ACM Elections Committee, c/o Pat Ryan, COO, ACM Headquarters, 2 Penn Plaza, Suite 701, New York, NY 10121, USA, by January 31, 2014. Duplicate copies of the petitions should also be sent to the Chair of the Elections Committee, Gerry Segal, c/o ACM Headquarters. All candidates nominated by petition are reminded of the requirements stated in the Policy and Procedures on Nominations and Elections that a candidate for high office must meet in order to serve with distinction. This document is available on: http://www.acm.org/about/acm-policies-procedures, or copies may be obtained from Rosemary McGuinness, Office of Policy and Administration, ACM Headquarters. Statements and biographical sketches of all candidates will appear in the May 2014 issue of *Communications of the ACM*.

The Nominating Committee would like to thank all those who helped us with their suggestions and advice.

Alain Chesnais, CHAIR,
Sheila Anand, Susan Dumais, Ben Fried, and Fabrizio Gagliardi

DOI:10.1145/2541883.2541887

ACM's FY13 Annual Report

FY '13 was an exceptional year for ACM. Membership reached a record high for the 11th consecutive year, solidifying ACM's lead as the world's largest educational

and scientific society in computing. Recognizing the expressed desires of our membership and the research community at large, ACM took a bold new approach to—and stand on—open access publishing, one that will no doubt set the tone for other professional organizations. We witnessed the steadfast commitment and global impact of our hubs in Europe, India, and China. And we continue to enrich ACM's pledge to educate future generations about the wonders of computer science with new initiatives and joint efforts that share our resources with the world.

Last April, ACM ushered in several publishing policy changes that increased access to its journals and conference proceedings. The changes were designed to balance the needs of authors and researchers in the computing community by expanding author rights as well as enabling SIGs to sponsor the open access to their most current proceedings. These latest policy changes serve as another step forward in an ongoing process in which ACM adapts to the new realities of scholarly publishing and prepares for an open access future.

ACM's international initiatives continue to flourish with ACM Europe, ACM India, and ACM China working to build a greater following in these territories by spreading the word and sharing the resources with technology associations and educators worldwide. Indeed, ACM Europe was officially incorporated as a legal entity in Europe this June—a remarkable feat for an organization less than four years old. With this status, ACM Europe can participate in discussions with the European Union on such topics as computing research, technology policies, and education priorities.

Education is the heartbeat of ACM: be it steering the computing curriculum for students and educators; taking the lead in equipping K–12 teachers with the tools and talent to teach next generations; providing publications of the highest quality to nourish today's professionals and scholars; or advising policymakers on the merits of computing as a core component to a student's future. I was proud to be part of the first Heidelberg Laureate Forum (HLF)—an event aimed at broadening the vision of young researchers in computer science and mathematics by connecting them with many of the preeminent scientists in the field. The inaugural HLF gathered more than 25 ACM A.M. Turing Award recipients and winners of other prestigious honors to share information and insights with 200 young researchers from around the world. Imagine the young student being able to sit down with the very role models that sparked their computing passions. It was an extraordinary event—a natural for ACM—and I look forward to next year's forum.

We continue to enrich ACM's pledge to educate future generations about the wonders of computer science.

The following report lists just some of the many activities and accomplishments of the Association over the fiscal year. As I write this letter, we are deep into the first quarter of FY14, with many more plans and challenges to address. In many ways, ACM is at a turning point. With the open access movement changing community expectations about how publications should be financed and distributed; with our international presence thriving but SIG membership declining, we must look to restructure ACM's business models in order to build a robust interconnected set of current and future activities, programs, and products. This challenge is of paramount importance in the coming year. Indeed, senior ACM leaders and staff recently held a two-day retreat to sort through the options and explore new ways of thinking about products and services.

As always, we look to our devoted volunteers and members to share their insights and ideas with us. Together, we can prepare ACM for a future even more accomplished and amazing than its past.

Vinton G. Cerf, ACM PRESIDENT

Highlights of ACM Activities: July 1, 2012–June 30, 2013

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

ACM leadership, along with the ACM Publications Board, responded to appeals to make the association's scholarly articles more openly accessible by spearheading a comprehensive review of its copyright policy. These efforts resulted in major changes to ACM's publishing rights model that were introduced last April. ACM authors exercise greater control of their published works as they are now offered three choices for managing rights: an author-pays open access option, an exclusive license agreement, along with the traditional copyright transfer. In addition, ACM took steps to allow SIGs to open up more of their conference content.

The centerpiece of ACM publications is the ACM Digital Library (DL) serving as the primary distribution mechanism for all the association's publications as well as host to scientific periodicals and a set of conference proceedings from external organizations. The DL, now available at 2,650 institutions in 64 countries, boasts an estimated 1.5 million users worldwide. The result of this widespread availability led to more than 15 million full-text downloads in FY13.

ACM is committed to increasing the scope of material available via the DL. Last year, over 30,000 full-text articles were added, bringing total DL holdings to 380,000 articles. ACM's *Guide to Computing Literature* is also integrated within the DL. More than 150,000 works were added to the bibliographic database in FY13, bringing the total *Guide* coverage to more than 2.2 million works.

ACM is the publisher of 79 periodicals, including 41 journals and transactions, eight magazines, and 30 newsletters as of year-end FY13. During the year, ACM added 465 volumes of conference and related workshop proceedings to its portfolio. In addition, a collection of over 1,260 e-books is now assimilated into the DL, available to all ACM members. Moreover, the ACM International Conference Proceedings Series (ICPS) added 102 new volumes, a significant increase over FY12.

A proposal to reinstate the ACM Press Book Series was adopted by the ACM Publications Board, 12 years after the previous book series effort was discontinued. This new series, in partnership with Morgan & Claypool publishers, will take advantage of the opportunities presented in the scholarly e-book market and focus primarily on academic-oriented research monographs and graduate-level textbooks with an emphasis on unique, innovative works.

The Publications Board's initiative to update the 1998 Computer Classification System was finalized in FY13. It was an exhaustive effort involving 160 domain experts and 13 subdiscipline-specific teams. The previous CCS terms are now mapped to the new version and all articles appearing in the DL reflect the new CSS concepts.

The Digital Library, now available at 2,650 institutions in 64 countries, boasts an estimated 1.5 million users worldwide.

Transactions on Economics and Computation made its debut this year; next on deck is *Transactions on Spatial Algorithms and Systems* and *Transactions on Parallel Computing* coming later this year.

Education

ACM continues to lead the computer science education community through the work of the ACM Education Board, the ACM Education Council, ACM SIGCSE, Computer Science Teachers Association (CSTA), and ACM Education Policy committee.

ACM's Education Board readied results from its first survey for non-doctoral-granting institutions in computing (NDC). The goal of this annual report is to help fill the gaps in data on non-Taulbee programs and contribute a more complete view of the academic landscape in computing. Indeed, the report confirmed positive trends in enrollment and degree production at participating not-for-profit U.S. academic institutions that grant bachelor's and/or master's degrees in major computing disciplines.

A white paper addressing the growing popularity of massive open online courses (MOOCs) was published this year that outlined the challenges and opportunities presented by new technologies and current educational experiments. The work was a combined effort between the ACM Education Board, Council, and CSTA.

A 10-year effort to revise and revitalize the computer science curriculum guidelines was finalized this year with the release of the ACM/IEEE-CS Computer Science Curriculum (CS2013). Several high-level themes provided an overarching guide for the development of CS2013, including the importance of viewing CS as a discipline actively seeking to work with other disciplines; reevaluating the essential topics with enough flexibility to add new ones as needed; identifying existing exemplar courses; and understanding that curri-

cula exists within specific institutional needs, goals, and constraints.

The CSTA continues to thrive as a key component in ACM's efforts to see real computer science exist and count at the high school level. CSTA membership increased 27% to a record 13,966 in FY13. The organization released four pivotal reports that examine the state of computer science in the K–12 environment, be it CS teacher readiness, student experiences, or major education research projects.

Under the guidance of the Education Policy Committee, ACM continued its efforts to reshape the U.S. education system to see real computer science exist and count as a core graduation credit in U.S. high schools. Working with the CSTA, the National Center for Women and Information Technology, NSF, Microsoft, and Google, ACM helped launch a new public/private partnership under the leadership of Code.org to strengthen high school level computing courses, improve teacher training, engage states in bringing computer science into their core curriculum guidelines, and encourage more explicit federal recognition of computer science as a key discipline in STEM discussions.

Several SIGs hosted innovative educational programs and special projects throughout the year. For example, one of the major objectives of the ACM SIGGRAPH Education Committee is to help establish a worldwide network of computer graphics educators. This year, the committee's initiatives were promoted at conferences in Brazil, Germany, Spain, and Mexico.

Professional Development

The Practitioners Board and Professional Development Committee (PDC) directed many new products and initiatives designed for computing professionals and managers.

The ACM PDC continued building on the success of the previous year, notably increasing the frequency and reach of its webinar program. The committee added six webinars in FY13, bringing the total number to 10. Among the featured topics: the future of the Internet; engineering SaaS; IBM Watson; and parallel computing.

The PDC also worked to tighten the integration of ACM assets such as the addition of Turing Centenary videos, *Queue* content, and other ACM learning tools like Tech Packs and podcasts in the ACM Learning Center—the hub of ACM's learning ecosystem.

The Practitoners Board and Professional Development Committee directed many new products and initiatives designed for computing professionals and managers.

ACM Queue, the online practitioner's magazine spirited by the Practitioner Board, again surpassed the million-pageview threshold, with 1,039,447 pages viewed over the last 12 months.

ACM Queue also began creating and publishing video portraits—interviews with young practitioners who are ACM members. Five portraits were published this year and the response showed great promise.

Public Policy

ACM's U.S. Public Policy Council (USACM) educates policymakers in many areas of potential legislation. This year the committee provided support, voiced opposition, or gave expert feedback to lawmakers regarding the Identity Ecosystem Steering Group, the continuity of e-government, reform of the Electronic Communications Privacy Act and the Computer Fraud and Abuse Act, and noted concerns about federal caps on scientific and technical conference spending. USACM also offered advice on how the U.S. Patent and Trademark Office and the software community could enhance the quality of software-relat-

ed patents and outlined ways in which technology issues should be treated in the context of voting systems and election reform.

The Committee on Computers and Public Policy assists ACM in a variety of internationally relevant issues pertaining to computers and public policy. CCPP's respected *ACM Forum on Risks to the Public in Computers and Related Systems*, designed to discuss potential and serious computer-related risks with a global audience, covers such issues as human safety, privacy, election integrity, and societal/legal responsibilities.

The Education Policy Committee and CSTA remains deeply involved in Computing in the Core coalition efforts to increase access to rigorous computing courses for all students. In the last year, CSTA members participated in many events promoting education initiatives on this front and met with legislators to discuss STEM education issues.

The Committee on Professional Ethics (COPE) engages in a variety of projects to promote professionalism and ethical behavior consistent with and supportive of the ACM's position on professional ethics. Along with crafting workshops devoted to methods of teaching ethics and decision making, COPE endorsed the Pledge of the Computing Professional and worked with other computer societies nurturing ethics.

SIGCHI's International Public Policy Committee is finalizing a report to serve as a foundation for the topic of human-computer interaction and public policy.

Balance Sheet: June 30, 2013 (in Thousands)

ASSETS	
Cash and cash equivalents	$35,237
Investments	69,608
Accounts receivable and other current assets	5,766
Deferred conference expenses and other assets	6,045
Fixed assets, net of accumulated depreciation and amortization	983
Total Assets	**$117,639**

LIABILITIES AND NET ASSETS	
Liabilities:	
Accounts payable, accrued expenses, and other liabilities	$10,442
Unearned conference, membership, and subscription revenue	24,683
Total liabilities	**$35,125**
Net assets:	
Unrestricted	75,800
Temporarily restricted	6,714
Total net assets	**82,514**
Total liabilities and net assets	**$117,639**

Optional Contributions Fund — Program Expense ($000)	
Education Board accreditation	$95
USACM Committee	20
Total expenses	**$115**

Students

The 37th Annual ACM International Collegiate Programming Contest (ACM-ICPC) took place in St. Petersburg, Russia, with 120 teams competing in the World Finals. Earlier rounds of the competition included nearly 30,000 contestants representing 2,300 universities from 91 countries. Financial and systems support for ICPC is provided by IBM. The top four teams won gold medals as well as employment or internship offers from IBM.

The ACM Student Research Competition (SRC), sponsored by Microsoft Research, continues to offer a unique forum for undergraduate and graduate students to present their original research at well-known ACM-sponsored and co-sponsored conferences before a panel of judges and attendees. This year's SRC saw graduate and undergraduate winners compete against more than 219 participants in contests held at 17 ACM conferences.

The ACM-W Scholarship program further enhanced its support for women undergraduate and graduate students in CS and related programs. The committee awarded 33 student scholarships in FY13 to students to attend research conferences around the world.

SIGPLAN's mentoring initiatives are designed to encourage and support the next generation of the programming languages community. Like all ACM SIGs, SIGPLAN supplies financial support for students to attend conferences. Moreover, the group supports a Programming Languages Mentoring workshop to encourage students to pursue careers in this field.

Internationalization

ACM Europe was incorporated into a legal entity in FY13. As a legal entity, ACM Europe is now able to engage and influence EU-wide policy and par-

Statement of Activities: Year ended June 30, 2013 (in Thousands)

REVENUE	Unrestricted	Temporarily Restricted	Total
Membership dues	$8,574		$8,574
Publications	20,207		20,207
Conferences and other meetings	25,424		25,424
Interests and dividends	1,854		1,854
Net appreciation of investments	2,805		2,805
Contributions and grants	3,634	$1,689	5,323
Other revenue	234		234
Net assets released from restrictions	1,894	(1,894)	0
Total Revenue	**64,626**	**(205)**	**64,421**
EXPENSES			
Program:			
Membership processing and services	$807		$807
Publications	11,390		11,390
Conferences and other meetings	23,206		23,206
Program support and other	9,048		9,048
Total	**44,451**		**44,451**
Supporting services:			
General administration	10,743		10,743
Marketing	1,538		1,538
Total	**12,281**		**12,281**
Total expenses	**56,732**		**56,732**
Increase (decrease) in net assets	7,894	(205)	7,689
Net assets at the beginning of the year	67,906	6,919	74,825
Net assets at the end of the year	**$75,800**	**$6,714**	**$82,514***

* Includes SIG Fund balance of $36,793K

ticipate in discussions in areas such as technology, computing research, and education.

The first European Federated Research Conference took place in Paris last May, a banner event that incorporated five conferences on diverse aspects of computing. The meeting was a significant success for a first conference held during a difficult financial climate in Europe.

Last April, ACM India hosted the first-of-its-kind celebration of Women in Computing in India, providing a unique opportunity for collective learning and showcasing work in progress for many young scholars. This landmark event was designed to help empower women in computing in India, offering workshops and seminars focused on entrepreneurship, potential career paths, and best practices for women employees working in IT firms.

A faculty summit held last February by ACM India in cooperation with Microsoft Research India was a great success. "Scaling Up Research and Innovation in Indian Institutions" focused on education challenges in India, particularly issues facing young CS researchers and the quality and impact of MOOCs. In addition, ACM India established a new Education

2012 ACM Award Recipients

ACM A.M. TURING AWARD
Shafi Goldwasser and Silvio Micali

ACM-INFOSYS FOUNDATION AWARD IN THE COMPUTING SCIENCES
Jeffrey Dean and Sanjay Ghemawat

ACM/AAAI ALLEN NEWELL AWARD
Yoav Shoham
Moshe Tennenholtz

THE 2013–2014 ACM-W ATHENA LECTURER AWARD
Katherine Yelick

GRACE MURRAY HOPPER AWARD
Martin Casado
Dina Katabi

EUGENE L. LAWLER AWARD FOR HUMANITARIAN CONTRIBUTIONS WITHIN COMPUTER SCIENCE AND INFORMATICS
Thomas Bartoschek
Johannes Schöning

ACM-IEEE CS 2013 ECKERT-MAUCHLY AWARD
James Goodman

KARL V. KARLSTROM OUTSTANDING EDUCATOR AWARD
Eric Roberts

OUTSTANDING CONTRIBUTION TO ACM AWARD
Zvi Kedem

DISTINGUISHED SERVICE AWARD
Mateo Valero

PARIS KANELLAKIS THEORY AND PRACTICE AWARD
Andrei Broder
Moses Charikar
Piotr Indyk

SOFTWARE SYSTEM AWARD
LLVM
Chris Lattner
Vikram Adve
Evan Cheng

ACM-IEEE CS KEN KENNEDY AWARD
Mary Lou Soffa

DOCTORAL DISSERTATION AWARD
Shvamnath Gollakota

HONORABLE MENTION
Peter Hawkins
Gregory Valiant

ACM INDIA DOCTORAL DISSERTATION AWARD
Ruta Mehta

HONORABLE MENTION
Srikanth Srinivasan

Committee to consider how ACM can contribute to the growth of high-quality education throughout the region.

ACM China finished its third year of operation by extending its outreach efforts to bring ACM awareness to academic institutions and industries in this vast region. With each passing year the number of conferences, chapters, and memberships established in the region increases. ACM China also continues to explore partnership opportunities with the China Computer Federation (CFF), indeed for the last two years selected articles from *Communications of the ACM* have been translated into Chinese and published in CFF's magazine.

An important role for the Education Board is to improve understanding of the computer education landscape on a global scale. Over the last year, the board continued to collaborate with ACM Europe and Informatics in Europe by offering guidance on computing in schools throughout Europe.

International activities and conferences hosted by ACM SIGs increase every year. KDD-2012 was a huge success in Beijing, where its closing panel session on big data drew record-breaking attendance. SIGevo held a workshop at the University of Adelaide in Australia and its GECCO conference is slated for Amsterdam. And SIGAPP held its annual Symposium on Applied Computing conference in Coimbra, Portugal.

Electronic Community

ACM rolled out a new online e-Rights Transfer application system in April, giving authors new options for managing rights and permissions. The system, now used by all ACM journals, proceedings, and magazines, completely automates the rights transfer process.

The Technology and Tools Task Force developed a Web 2.0 website, Technology that Educators of Computing Hail (TECH). This effort is a reflection of its charter to promote great teaching by providing the best technology and tools resources for computing educators.

By the end of the year, ACM magazines *Communications of the ACM*, *ACM Inroads*, and *interactions* became accessible as easy-to-use mobile apps for iPhones, iPads, and Android devices. These new downloadable apps enable members to access their favorite ACM magazines in a new way.

ACM rolled out a new online e-Rights Transfer application system, giving authors new options for managing rights and permissions.

The Publications Board spirited a project to develop a webpage template to be used for all ACM journals and transactions. The result will give ACM publications a professional, uniform look and feel that enhances the imprint of one of ACM's top services—journal publishing.

ACM SIGs across the board continue to strengthen their online presence to build global awareness as well as incorporate social media into their operation at every opportunity. SIGACCESS, for example, enhanced its website with resources such as a set of guidelines that reflect current thinking on language for writing in the academic accessibility community as well as a guide for planning accessible conferences. The conference program for SIGUCCS 2012 was offered via a mobile app that allowed attendees to view the program, choose their sessions, and submit session evaluations using their mobile devices. SIGMOD's official blog catches the heartbeat of the community on exciting and controversial topics from posts by notable researchers and teachers in the database community. The SIG's DBJobs online service continues to attract job seekers with a database background.

Conferences

SIGGRAPH 2012 welcomed 21,212 artists, research scientists, gaming experts and developers, filmmakers, students, and academics from 83 countries to Los Angeles. More than 1,200 speakers and contributors participated in the event and SIGGRAPH's exhibition hall drew 161 industry organizations from 19 countries.

KDD-2012 attracted a record high in attendance and the number of paper submissions. In addition, the conference introduces an option for every selected paper to be accompanied by a 30-second video summary, an Asia-Pacific track, and an industry practice expo that resulted in standing-room-only attendance.

The flagship conference for the ACM Special Interest Group on Data Communication (SIGCOMM) continues to thrive in scope and attendance. This year's conference, held in Helsinki, drew over 600 attendees.

Recognition

There were 125 new chapters chartered in FY13. Of the 15 new professional chapters, all were internationally based; of the 110 new student chapters, 60 were international.

The ACM Fellows Program recognized 52 members for their contributions to computing and computer science in FY13. The new inductees brought the number of ACM Fellows to over 750.

ACM also named 41 new Distinguished Members in FY12, of which there were six Distinguished Educators, three Distinguished Engineers, and 32 Distinguished Scientists, bringing the total number of Distinguished Members to 326.

ACM-W's Regional Celebration Committee provided support for women in computing events in Australasia, Chicago, Kentucky, New Mexico, New York, Nova Scotia, Ohio, Ontario, the Pacific Northwest, Pune (India), and the Rocky Mountain region. In addition, ACM continues to partner with the Anita Borg Institute in presenting the annual Grace Hopper Celebration of Women in Computing.

tvx 2014

ACM INTERNATIONAL CONFERENCE ON INTERACTIVE EXPERIENCES FOR TELEVISION AND ONLINE VIDEO

25-27 JUNE, 2014
NEWCASTLE UPON TYNE
UK

Paper Submissions by
3 February 2014

Workshop, Demo, WIP
DC, Grand Challenge
& Industrial
Submissions by
31 March 2014

Welcoming Submissions on

Content Production
Systems & Infrastructures
Devices & Interaction Techniques
Experience Design & Evaluation
Media Studies
Data Science & Recommendations
Business Models & Marketing
Innovative Concepts & Media Art

TVX2014.COM

DOI:10.1145/2541883.2541886

U.S. Does Not Control the Internet

In his editor's letter "The End of The American Network" (Nov. 2013), Moshe Y. Vardi made this startling statement: "Thus, in spite of its being a globally distributed system, the Internet is ultimately controlled by the U.S. government. This enables the U.S. government to conduct Internet surveillance operations that would have been impossible without this degree of control." This is untrue on several levels: First, so-called U.S. control of the Internet is limited to approval of root-zone changes of the Domain Name System, though the U.S. has never exercised that authority against any top-level delegation or re-delegation proposed by the Internet Corporation for Assigned Names and Numbers (http://www.icann.org), the not-for-profit organization that oversees the Internet's naming and numbering system. In addition, root-zone servers exist outside the U.S., and any heavy-handed attempt by the U.S. government to exercise unwarranted control over the contents of the zone would be international political suicide and likely cause a near immediate takeover by operators in other countries. Second, this administrative function has nothing to do with the routing of information on the Internet and does not provide any agency of the U.S. government any advantage for surveillance of Internet traffic.

Although the topology of the early Internet was such that much of the world's traffic flowed through the U.S., it was a historical artifact of the Internet's early development. More recently, the pattern changed radically, with Internet topology evolving into a more comprehensive global mesh structure.

Vardi's repetition of spurious and incorrect claims, often made for political reasons by other countries, gives credence to ignorance while illustrating the extent to which a knee-jerk reaction generated by Edward Snowden's recent disclosures concerning the National Security Agency's surveillance of personal communications worldwide has been unthinkingly adopted by otherwise presumably sensible individuals. A retraction of Vardi's statement is essential to confirm ACM is a professional organization, not a thoughtless echo chamber for uninformed sentiment.

George Sadowsky, Woodstock, VT
(The author is a member of the ICANN Board of Directors)

Author's Response:

I am not an Internet expert and am happy to be educated by Internet insiders like Sadowsky. But the revelations that have poured out for the past many months resulted in a massive loss of public trust in insiders. It behooves Internet insiders like Sadowsky to speak up and explain precisely what role the U.S. government plays in Internet governance and what has enabled the massive Internet surveillance operations run by the National Security Agency. Only more transparency, rather than vehement denials, may begin the process of rebuilding the public's trust in insiders.

Moshe Y. Vardi, Editor-in-Chief

Just the Facts... for Ethics Sake

Katina Michael's and MG Michael's "Computer Ethics" column "No Limits to Watching?" (Nov. 2013) was marred by a careless discussion of HeLa, an immortal line of human-derived cells that is today an important tool for biomedical research since being derived from a sample of cervical cancer cells taken from Henrietta Lacks, a patient at Johns Hopkins Hospital in Baltimore, in 1951.

The Michaels said Henrietta Lacks's "...cells were 'taken without her knowledge.'" In fact, she had a biopsy, like millions of other cancer patients. They further said, "Until this year... HeLa cells were 'bought and sold...' without compensation." The understanding agreed in August between the National Institutes of Health and Lacks's family was, in fact, about access to genomic data, not compensation. Lacks's granddaughter Jeri Lacks Whye even said to the NIH: "The Lacks family is honored to be part of an important agreement that we believe will be beneficial to everyone."

The oncologists treating Lacks should indeed have asked her whether they could reuse her cells for research. But monetary payment in such cases could lead toward a market in human body parts. Your body is "yours" in many senses of the word, but not in the sense that you may sell it. This is a consequence of the 13th Amendment to the U.S. Constitution and other anti-slavery laws around the world.

The Michaels further said, "Consider the story of Henrietta Lacks, whom scientists named 'HeLa.'" In my experience, even this is inaccurate. I have heard many scientists say they work with "HeLa cells," as well as with BL321 cells and CHO DG44 cells not of human origin. When I have heard them speak of Henrietta Lacks, they have called her by her full name.

The Michaels rightly said, "There is a stark asymmetry between those who use wearables and those who do not. ... Maker or hacker communities ... create personalized devices ... [which] often [are] commercialized for mass consumption." I suspect they were trying to reinforce this point by saying if "scientists" disrespected Henrietta Lacks, then perhaps engineers devising wearable devices today would likewise be disrespectful of bystanders in the field of view.

Such lazy generalizations about scientists can hardly help readers like me address the ethical challenges we face.

Chris Morris, Chester, U.K.

Planned [Software] Immortality

I could not disagree more with Marshall Van Alstyne's column paean to mortality "Why Not Immortality?" (Nov. 2013) because, unlike the living creatures and appliances he modeled, there is no aspect of software that cannot be designed to evolve. Software becomes obsolete and must be replaced only if its designers do not design for immortality. Consider our cities. Al-

though Julius Caesar would not recognize Paris or Rome today as the places he knew, they existed long before his time, and will continue to exist long after mine. We do not abandon cities and replace them with new ones; cities live forever because we continuously reinvent and renew them. Over time, neighborhoods built for horses and wagons have evolved to suit cars. Office parks replace factories. But even as they morph, cities live on, except in the rare instance when one is destroyed by natural disaster or by war. Although nothing short of the universe itself is truly immortal—and even that is in doubt—it is time to stop viewing software through the automobile designer's mind-set of planned obsolescence and view it instead through the urban planner's mind-set. Great software should last forever but can do so only if its designers think of it that way, investing in continuous reengineering to make it happen.

K.S. Bhaskar, Malvern, PA

Cyberstalking Already Addressed

Esther Shein's news item "Ephemeral Data" (Sept. 2013) said the ex-boyfriend of a woman in Florida had posted nude photos of her online and that the woman now wanted a law to "criminalize 'cyberstalking'" in Florida. I would like to point out this would be a waste of her time since Florida already has a cyberstalking law under state statute 784.048 subsection 1.d. Moreover, I do not see how enacting such a law has anything to do with being stalked in the first place since the legal definition of stalking involves (as defined by the Florida Legislature) "...willful, malicious harassment or following evinced by a course of conduct over a period of time, no matter how short, including any acts of credible threats to one's safety, the safety of those close to that person, or the safety of the person's family members."

Since the photos of the Florida woman allegedly exist, one can assume she consented to them being taken. If not, another Florida law comes into play concerning voyeurism. If the woman indeed consented to the photos, she should have thought about having them taken in the first place, since we all know relationships do not necessarily last forever. By the way, nude photos do not constitute porn.

Anthony Cunarro, Palm Beach, FL

Ismail Cem Bakir Freed from Jail

My letter "Free Ismail Cem Bakir" (Oct. 2013) concerned Istanbul Technical University computer student Ismail Cem Bakir, who had been arrested and illegally imprisoned by the Turkish government in July following anti-government protests in June. I am now pleased to say he has been released, with all charges dropped. I sent the letter to my colleague, Vladimir Lifschitz, of the University of Texas, who was to lecture at the 29th International Conference on Logic Programming in Istanbul, August 24–29, and who then took time to speak on this violation of Bakir's human rights.

Several weeks later, a computer scientist, fluent in Turkish and who attended Lifschitz's lecture, informed him the Turkish government had indeed released Bakir, finding this information in a search of the Turkish website "ITU Gezi Forum #8"; Gezi is the park in Istanbul that was at the center of anti-government protests.

The translation he sent said 29 people, mostly university students, including Bakir, had been arrested and held illegally for four days, quoting Bakir saying "...he was in the stands to watch the graduation ceremony on July 8, and that he noticed plain-clothes police officers, summoned by the administration, taking photos of him and of many others like him. He said that might be one of the reasons he was taken into custody. He thanked his professors and friends who didn't leave him alone during his time in custody and in the Çağlayan courthouse."

Publicizing information on and support for our colleagues can be critical when their scientific freedom and human rights are at risk. As seen from Bakir's statement, his colleagues helped his cause and improved his morale while doing no harm. Publicity in *Communications* and other journals is also extremely useful.

Jack Minker, College Park, MD

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

BLOG@CACM

The *Communications* Web site, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter
Follow us on Twitter at http://twitter.com/blogCACM

DOI:10.1145/2555813 http://cacm.acm.org/blogs/blog-cacm

MOOCs Need More Work; So Do CS Graduates

Mark Guzdial assesses the first full year of massive open online courses, while Joel C. Adams considers the employment outlook for CS grads.

Mark Guzdial
"Results from the First-Year Course MOOCs: Not There Yet"
http://bit.ly/1gX4AYl
October 18, 2013

MOOCs in the Coursera, Udacity, and edX form are tightly tied to CS. The leaders of the xMOOC movement came out of computer science, and most of the first generation of xMOOCs focused on teaching computer science. Many of the MOOC evaluations so far have been expert reviews. Our Learning Sciences and Technologies seminar at Georgia Tech's College of Computing just read Moti Ben-Ari's travelogue on his experiences in Coursera's and Udacity's introductory CS MOOC. The empirical results of the first rounds of MOOCs on intro courses are now in, so it is worth considering how they are doing.

Karen Head has finished her series of posts in *The Chronicle of Higher Education* on the freshman-composition MOOC she taught with Gates Foundation funding. The stats were disappointing—only 238 of the approximately 15K students who did the first homework finished the course. That is even less than the ~10% we saw completing other MOOCs.

Karen Head writes:

No, the course was not a success. Of course, the data are problematic: Many people have observed that MOOCs often have terrible retention rates, but is retention an accurate measure of success? We had 21,934 students enrolled, 14,771 of whom were active in the course. Our 26 lecture videos were viewed 95,631 times. Students submitted work for evaluation 2,942 times and completed 19,571 peer assessments (the means by which their writing was evaluated). However, only 238 students received a completion certificate—meaning that they completed all assignments and received satisfactory scores.

Our team is now investigating why so few students completed the course, but we have some hypotheses. For one thing, students who did not complete all three major assignments could not pass the course. Many struggled with technology, especially in the final assignment, in which they were asked to create a video presentation based on a personal philosophy or belief. Some students, for privacy and cultural reasons, chose not to complete that assignment, even when we changed the guidelines to require only an audio presentation with visual elements. There were other students who joined the course after the second week; we cautioned them that they would not be able to pass it because there was no mechanism for doing peer review after an assignment's due date had passed.

Georgia Tech also received funding from the Gates Foundation to develop a MOOC approach for a first-year college physics course. I met with Mike Schatz, the lead teacher on that effort. It is a remarkable course, including a "laboratory" where students take videos of moving objects, then construct computational simulations in Python to match the real-world observations. The completion results were pretty similar to Karen's: 20K students signed up, 3K students completed the first assignment, and only 170 finished.

In terms of empirical studies, Mike had an advantage that Karen did not — there are standardized tests for measuring the physics knowledge he was testing, and he used those tests before and after the course. Mike said the completers fell into three categories: those who came in with a lot of physics knowledge and who ended with relatively little gain, those who came in with very little knowledge and made almost no progress, and a group of students who really did learn a lot. They do not yet know the relative percentages of the three categories. However, it is clear that being a *completer* does not mean that anything was *learned*.

I also met with Jason Freeman who finished his Survey of Music Technology MOOC for Coursera. His results

were a bit better: 24K signed up, 13K visited, and 900 completed. It seems the more advanced the course, the better the completion rate.

The report from San Jose State University's MOOC experiment with a remedial mathematics course found: *The researchers say, perhaps unsurprisingly, that what mattered most was how hard students worked. "Measures of student effort trump all other variables tested for their relationships to student success, including demographic descriptions of the students, course subject matter, and student use of support services."*

It is not surprising, but it is relevant. Students need to make an effort to learn. New college students, especially first-generation college students, may not know how much effort is needed. Who will be most effective at communicating the message about effort and motivating that effort—a video of a professor, or an in-person professor who might even learn your name?

MOOC companies have set a goal of democratizing education. They aim to make education available to people who would not otherwise get access to education. MOOCs are not yet succeeding at that goal. The empirical findings (for example, Armando Fox's results at Berkeley and Tucker Balch's demographics) suggest MOOCs draw mostly men (especially in the CS MOOCs) who already hold degrees and are overwhelmingly from the U.S. and the developed world. Right now, xMOOCs seem most successful for professional and continuing education. Gary May, dean of engineering at Georgia Tech, recently wrote in *Inside Higher Ed*, "The prospect of MOOCs replacing the physical college campus for undergraduates is dubious at best. Other target audiences are likely better-suited for MOOCs." That summarizes the current state pretty well.

Joel C. Adams "Hot Job Market for Computer Science Graduates"

http://bit.ly/JaSMim
April 19, 2012

Back in 2010, companies were hiring computing graduates as fast as we could produce them, but there was a widespread misperception that U.S. computing jobs were in danger of being offshored. Many people still believe this.

Career	Projected Growth (New Jobs)	Percentage Of All STEM Growth
1. Software developer	314,600	30%
2. Systems analyst	120,400	12%
3. Computer support	110,000	11%
4. Network/system administrator	96,600	9%
5. Network architect, Web developer, computer security professional	65,700	6%

To counteract this, I put together a Web page called the *Market for Computing Careers*. My basic idea was to create visualizations of the U.S. Bureau of Labor Statistics (US-BLS) employment projections and data on bachelor's degrees awarded, to help people—especially students, parents, teachers, and guidance counselors—understand what the U.S. government was predicting the job market for CS graduates would be like. I had seen fragments of this data reported in piecemeal fashion, but I wanted to collect these pieces in one place to try and tell a more complete story.

I updated this Market For Computing Careers page using the new US-BLS 2010–2020 employment projections, as well as the most recent U.S. STEM graduation data (2008) available from the National Science Foundation.

The new US-BLS projections predict the already hot job market for computing professionals will become even hotter this decade. Excluding health care, these projections predict the five careers in science, technology, engineering, and mathematics (STEM) with the most growth will *all* be in computing:

Taken collectively, these projections predict computing careers will make up 73% of the jobs in STEM careers this decade compared to 16% in (non-software) engineering, 9% in the natural sciences, and 2% in mathematical sciences.

To try to predict how competitive the job environment will be, we can combine the US-BLS total job projections (new jobs plus retirement-replacements) with graduation data from the NSF. Dividing the total projected computing jobs per year by the number of computing bachelor's degrees awarded in the most recent year yields a jobs/grads ratio of 3.5 computing jobs per person graduating with a bachelor's degree in computing. (In 2010, this computing jobs/grads ratio was 2.9.) By contrast, the total jobs/grads ratio is below 1.0 in every other STEM area.

This data suggests on average, there will be 97,000 more U.S. computing jobs than graduates each year, a shortfall that even the current H1B Visa Quota is insufficient to address. To meet this decade's demand with homegrown talent, U.S. colleges and universities would need to produce 3.5 times as many computing graduates per year as they did in 2008. The Taulbee Survey data has shown modest increases in computing graduation rates the past two years at Ph.D.-granting institutions, but the observed increases do not come close to addressing the projected demand.

Companies seeking U.S. computing professionals will thus be competing with other companies for a limited supply of personnel. We are already seeing this competition, as many of our students are receiving multiple internship offers, and many of our graduates are receiving multiple job offers. The US-BLS projections suggest this competition will likely increase this decade.

For visualizations of this data, see the 2012 Market For Computing Careers page at http://cs.calvin.edu/p/ComputingCareersMarket. For comparison purposes, see the 2010 Market For Computing Careers page at http://cs.calvin.edu/images/department/jobs/2018/.

Now is a great time to be a computing major as the abundance and variety of computing jobs over the next decade should make it relatively easy to find a career that is stimulating, fulfilling, and that compensates well. Help spread the word!

Mark Guzdial is a professor at the Georgia Institute of Technology. **Joel C. Adams** is a professor of computer science at Calvin College.

HEIDELBERG LAUREATE FORUM

Association for Computing Machinery

Inviting Young Scientists

Meet Some of the Greatest Minds of Mathematics and Computer Science

Young researchers in the fields of mathematics and/or computer science are invited to participate in an extraordinary opportunity to meet some of the preeminent scientists in the field. ACM has joined forces with the Heidelberg Laureate Forum (HLF) to bring students together with the very pioneering researchers who may have sparked their passion for science and math. These role models include recipients of the Abel Prize, the ACM A.M. Turing Award, and the Fields Medal.

The next Heidelberg Laureate Forum will take place September 21–26, 2014 in Heidelberg, Germany.

The week-long event will focus on scientific inspiration and exchange through a series of presentations, workshops, panel discussions, and social events involving both the laureates and the young scientists.

Who can participate?

The HLF invites new and recent Ph.D's, Ph.D. candidates, other graduate students involved in research and undergraduate students with solid experience in and a commitment to computing research to apply.

How to apply:

Young researchers can apply online:

https://application.heidelberg-laureate-forum.org/

The materials required for a complete application are listed on the site.

What is the schedule?

The deadline for applications is **February 28, 2014**.

We reserve the right to close the application website early should we receive more applications and nominations than our reviewers can handle.

Successful applicants will be notified by **April 15, 2014**.

Science | DOI:10.1145/2555807 Gary Anthes

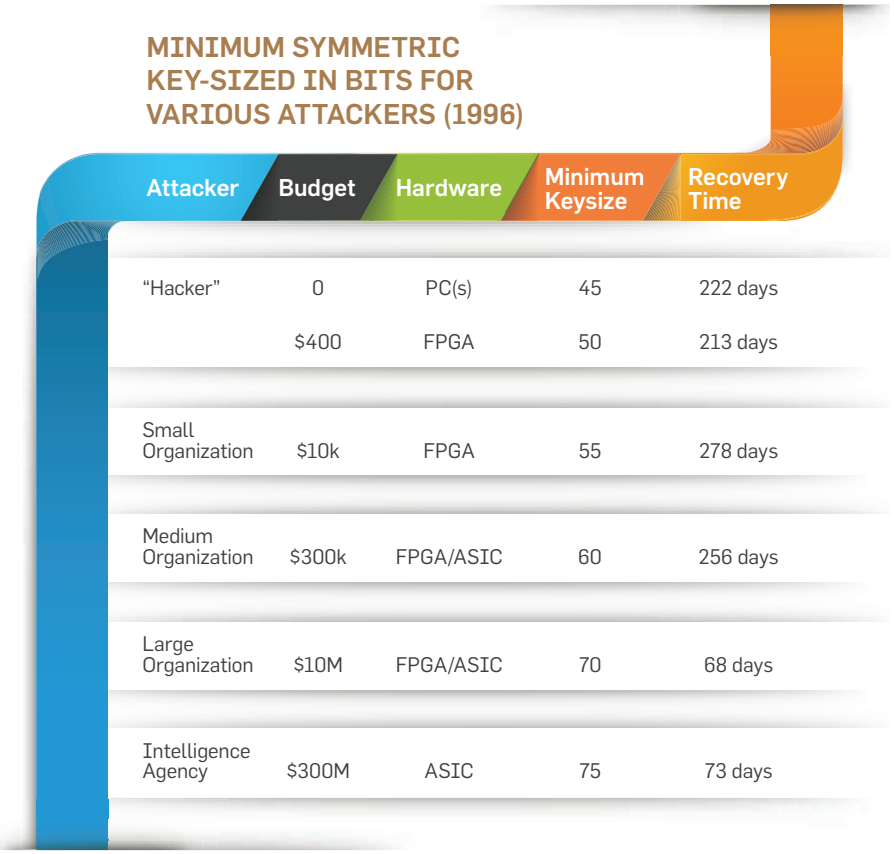
French Team Invents Faster Code-Breaking Algorithm

New method can crack certain cryptosystems far faster than earlier alternatives.

A team of French mathematicians and computer scientists has made an important advancement in the field of algorithms for breaking cryptographic codes. In a certain class of problem, the new algorithm is able to efficiently solve the discrete logarithm problem that underlies several important types of modern cryptosystems.

"Problem sizes, which did not seem even remotely accessible before, are now computable with reasonable resources," says Emmanuel Thomé, a researcher at the French Institute for Research in Computer Science and Control (INRIA) and one of four researchers reporting the advance. However, he notes, the new algorithm poses no immediate threat to most existing cryptosystems, including the RSA-based cryptography used in credit cards and much of e-commerce.

Virtually all the major types of cryptography in use today rely on the use of one-way functions, mathematical functions that are easy to compute but impractical to invert, or reverse. For example, it is easy to multiply togeth-



Cryptographic researcher Alfred Menezes says the new algorithm only becomes effective as the parameters of a cryptosystem, essentially the key size, grow asymptotically large.

er two large prime numbers, but it is computationally impractical to reverse that by factoring the resulting product. That is the basis of RSA cryptography. Similarly, it is easy to compute $a = x^n$ given x and n, but hard to compute the discrete logarithm n given a and x. The "discrete logarithm problem" is the basis for a number of cryptosystems, such as the Diffie-Hellman protocol and elliptic curve cryptography.

The "time complexity" of an algorithm is a measure of how execution time varies with input size, n. Those that work in "polynomial" time tend to be computationally feasible for large n, with execution time increasing in proportion to some polynomial with constant exponents, such as $5n^3 + 3n$. However, for algorithms of an "exponential" complexity, execution time varies with the nth power, so that for a sufficiently large n, the algorithm becomes computationally intractable. Computer scientists and mathematicians use a notation that varies from L(1), exponential, to L(0), polynomial. The advancement recently announced in essence moves the discrete logarithm problem from complexity L(1/3), sometimes called "sub-exponential," to a complexity close to L(0), or "quasi-polynomial."

What that means in practical terms depends on the exact nature of the problem and the input size, but in one example analyzed by the researchers, the new algorithm improves execution time from 2^{80} to 2^{70} operations, or about a factor of 1,000 faster. For that problem example, a computer configuration that would have taken three years to decipher a message before could now do it in a little more than a day, a feasible job for a sophisticated adversary. Also, the researchers point out, the algorithms in question exhibit asymptotic computational complexity, and the advantage of the new algorithm grows as the problem size increases.

"The computations that we have done recently would have been completely infeasible without this algorithm," says Antoine Joux, co-developer of the algorithm and a professor at the University of Versailles-Saint-Quentin-en-Yvelines. The latest algorithm is built on work by Joux in 2012, which moved the complexity down to L(1/4).

Yet as Joux points out, the new algorithm is not efficient against all discrete logarithms. The algorithm is intended for use on finite fields of small characteristic. The elements of a finite field can be expressed as polynomials, and for small characteristic finite fields, the coefficients of the polynomials are small integers. In a "binary finite field," for example, the coefficients are 0 and 1, and the characteristic is 2. However, for large characteristic finite fields, used for example in the digital signature algorithm (DSA), solving for the discrete logarithm remains a problem of sub-exponential complexity L(1/3).

Small characteristic finite fields

Milestones

Computer Science Awards, Appointments

SUTHERLAND RECEIVES KYOTO PRIZE FOR TECHNOLOGY

Ivan Sutherland, a Visiting Scientist at Portland State University, has been awarded the Kyoto Prize in Advanced Technology for "Pioneering Achievements in the Development of Computer Graphics and Interactive Interfaces."

Sutherland, who received the ACM A.M. Turing Award in 1988 and the IEEE John Von Neumann Medal in 1998, has been responsible for pioneering advances and fundamental contributions to computer graphics technology and interactive interfaces.

The Kyoto Prize honors those who have contributed significantly to the scientific, cultural, and spiritual betterment of mankind.

IEEE-CS HONORS TECHNICAL ACHIEVEMENT

The IEEE Computer Society recently honored five prominent technologists with Technical Achievement Awards in recognition of their contributions to the computing field.

The honorees include:

- **Jan Camenisch, research staff member and project leader at IBM Research-Zurich since 1999. Camenisch's research areas include public key cryptography, cryptographic protocols, practical secure distributed computation, and privacy-enhancing technologies.**
- **Virgil D. Gligor, a Carnegie Mellon University electrical and computing engineering professor and co-director of the University's CyLab. Gligor's research interests include access control mechanisms, penetration analysis, denial-of-service protection, cryptographic protocols, and applied cryptography.**
- **Kian-Lee Tan, a professor of computer science at the National University of Singapore. Tan's research interest in database systems focuses on query processing and optimization in a wide range of domains, including parallel, distributed, peer-to-peer, multimedia, high-dimensional, main memory, spatial-temporal, wireless, and mobile databases.**
- **Eva Tardos, Jacob Gould Schurman Professor of Computer Science and senior associate dean of Computing and Information Science at Cornell University. Tardos' research interest is algorithms and algorithmic game theory, the sub-area of theoretical computer science theory of designing systems and algorithms for selfish users.**
- **Philip S. Yu, a professor and Wexler Chair in computer science at the University of Illinois at Chicago. Yu's research is focused on big data, including data mining, data streams, databases, and privacy.**

GUGGENHEIM NAMES FATHER AND SON FELLOWS

Among the 175 scholars, artists, and scientists that recently received John Simon Guggenheim Memorial Foundation Fellowships on the basis of prior achievement and exceptional promise is a father-and-son team who were recognized in the natural Sciences field for their achievements in computer science.

Erik Demaine has been a professor of computer science at the Massachussetts Institute of Technology since 2001. His research interests involve various aspects of algorithms, from data structures for improving Web searches to the geometry of understanding how proteins fold, to the computational difficulty of playing games.

Martin Demaine has been the Angelika and Barton Weller Artist-in-Residence at the Massachusetts Institute of Technology since 2005. Martin works with his son Erik in paper, glass, and other materials; they use sculpture to help visualize and understand unsolved problems in mathematics.

Among their artistic works are curved origami sculptures in the permanent collections of the Museum of Modern Art in New York and the Renwick Gallery in the Smithsonian. Their scientific work includes more than 60 jointly published papers, including several about combining mathematics and art.

may be found in a type of cryptography known as "pairing-based," in which elements of two cryptographic groups are combined. This kind of cryptography is used in some systems that, for example, base encryption keys on personal attributes or the names of users. Pairing-based systems that use finite fields of small characteristic should now be avoided, the developers of the new algorithm say. "We are proving that some discrete log problems are much, much easier than we once thought," Thomé says. The warning may apply in particular to makers of hardware-based crypto, who like to use the small-characteristic, binary finite fields because they are easy to implement in hardware.

Some companies use pairing-based systems of their own design, but so far there is no firmly established standard for them, Thomé says. "They are an attractive option for applications like identity-based crypto or voting systems," he notes.

Alfred Menezes, who has studied the new algorithm as a cryptographic researcher at the University of Waterloo in Ontario, Canada, calls it "a fantastic algorithm—a stunning development." He says, "If I were a company today considering the use of pairing-based cryptography, I would be terrified of using small-characteristic pairings." In one case he studied, the algorithm succeeds in 2^{74} operations, vs. 2^{103} operations with the previous best algorithm. "While the 2^{74} computation is certainly a formidable challenge, with an organization like the NSA, it becomes feasible."

Menezes says his analysis also shows the algorithm only becomes effective as the parameters of a cryptosystem, essentially the key size, grow asymptotically large. It may not be efficient; in fact, it may be slower than other algorithms against systems of small key size. Also, it is not effective against RSA and other non-pairing-based systems.

The Algorithm

The new method lies in a family of what are called index calculus algorithms, used for computing discrete logarithms. It uses a "Las Vegas" algorithm, one that always yields the correct result, but in an amount of time that varies. Researchers can estimate runtimes, but cannot know exactly what the run time will be in any specific case. "We are sure of the result, but do not know if it will take two months or two months plus one week," says Razvan Barbulescu, a Ph.D. student at the University of Lorraine and a co-developer of the algorithm. However, they can be sure it would not take two minutes, he adds.

The new method lies in a family of what are called index calculus algorithms, used for computing discrete logarithms.

The algorithm is heuristic and involves the use of conjectures that cannot be proven. Says Barbulescu, "We can check experimentally that it works, but the math to prove it is difficult." Finally, he says, it is a recursive process, one that solves a big problem by solving smaller (and easier) instances of it.

The algorithm's recursion involves representing the elements of the finite field in a cascade of polynomials, starting with the polynomial from which the discrete log is to be computed. This is broken down into smaller polynomials, and those into successively smaller polynomials, into something called a "descent tree." Then, "building up an answer from the bottom with logs is pretty easy," says Barbulescu.

The breakthrough lay in finding a way to perform the descent process more efficiently than previously possible, he says, and in building a deeper tree with smaller problems at the end points.

The Future

Barbulescu says the research group has considered trying to push its ideas to medium- and large-characteristic systems, "but there is a huge difficulty porting this algorithm to these other cases," he says. "But if we were able to extend it to large characteristic, then it would be an earthquake in cryptography because every time there is an improvement in discrete logarithm, there is a corresponding improvement in factorization (RSA), because the problems are similar."

Meanwhile, though, existing RSA-based systems should be considered secure. "There are some buzz articles floating around on the Web saying that this is the endgame for RSA," Thomé says. "It is wrong to say that."

The University of Waterloo's Menezes says he is not aware of any cryptosystems in use today that are suddenly at risk because of the work by the French team. However, he warns, "There will be faster algorithms, better implementations of the existing algorithm perhaps through special-purpose hardware, and better analysis. Maybe the algorithms are faster than we think they are." C

Further Reading

Adj, Gora, Menezes, A., Oliveira, T., and Rodriguez-Henriquez, F.
Weakness of $F3^{6\cdot509}$ for discrete logarithm cryptography, http://eprint.iacr.org/2013/446 Presented at Crypto 2013 rump session, Santa Barbara, Calif., Aug. 20, 2013 http://crypto.2013.rump.cr.yp.to/.

Barbulescu, R., Gaudrey, P., Joux, A., and Thomé, E.
A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic, June 2013, preprint available at http://eprint.iacr.org/2013/400.pdf

Blake, I., Seroussi, G., and Smart, N.
Advances in elliptic curve cryptography, second edition, *London Mathematical Society Lecture Note Series*, April 2005 http://www.amazon.com/Advances-Elliptic-Cryptography-Mathematical-Society/dp/052160415X

Joux, A.
A new index calculus algorithm with complexity L(1/4 + o(1)) in very small characteristic, to be in Proceedings of Selected Areas in Cryptography 2013 (SAC 2013), Burnaby, British Columbia, Canada, August 2013. Paper at *Cryptology ePrint Archive*, Report 2013/095, 2013 http://eprint.iacr.org/2013/095

Menezes, A., van Oorschot, P., Vanstone, S.
Handbook of Applied Cryptography, CRC Press, October 1996, http://www.cacr.math.uwaterloo.ca/hac

Gary Anthes is a technology writer and editor based in Arlington, VA.

Technology | DOI:10.1145/2555809 Tom Geller

How Do You Feel? Your Computer Knows

Interfaces can sense your mood, if you let them.

Nearly a decade after its retirement, the advice-spewing "Clippy" remains one of technology's most hated characters. As part of Microsoft's Office Assistant help system, the paperclip-faced avatar proposed help based on Bayesian probability algorithms: start a word-processing document with "Dear," and it offered to help you write a letter. Express exasperation, and Clippy would gleefully continue pestering you: it could not sense your mood.

Perhaps Clippy would still be with us if it had employed affective computing, a growing field that attempts to determine a user's mood or emotion through visual, auditory, physiological, and behavioral cues—cues that comprise one's "affect." An affect-enabled Clippy might see your look of disgust and make itself scarce; conversely, it might pop up sooner when you furrow your brow in confusion.

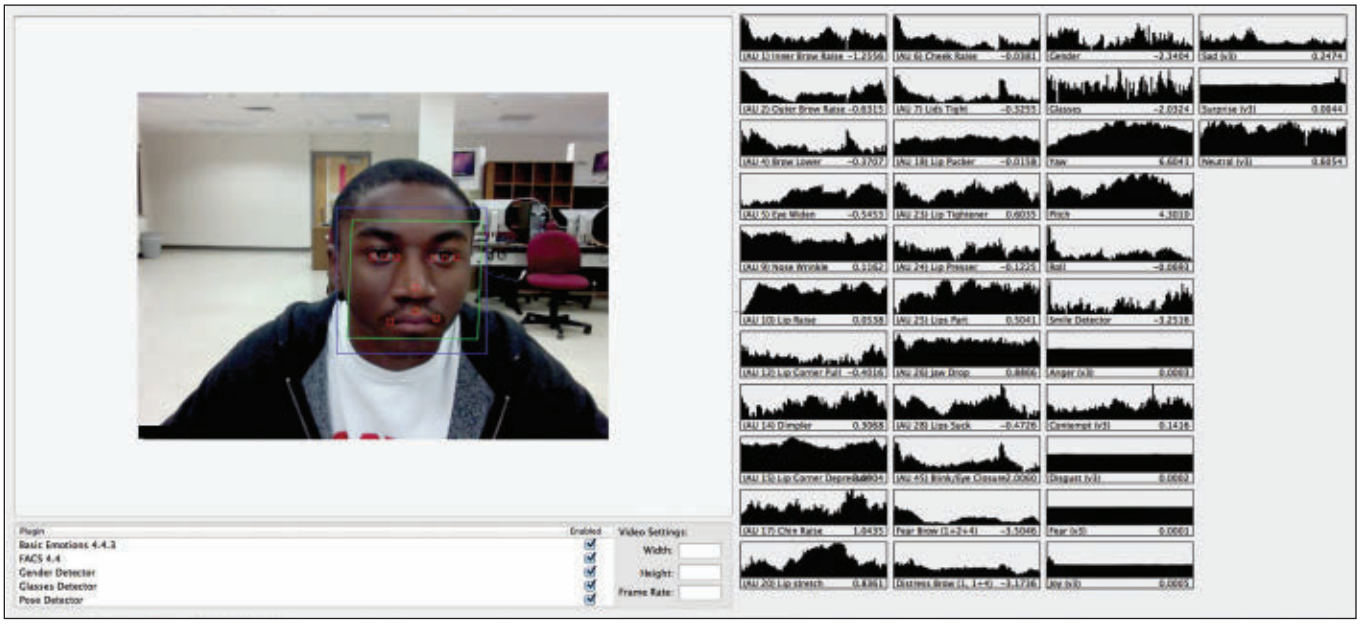
Affective computing could go well beyond desktop help systems. Researchers cite possibilities for educational, medical, and marketing applications, and have begun to commercialize their efforts. They are building systems that attempt to measure both emotions—short-term, responsive phenomena—and longer-term moods. Even as they surmount technical barriers, they will have to face ethical questions as they help machines access a formerly hidden area of your mind.

Deconstructing Moods and Emotions

Scientific interest in how mood and emotion affect interaction appears to be universal: Sun Tzu referred to "the art of studying moods" for military strategy in his sixth-century-BC classic *The Art of War*. However, it was not until the 19th century that empiricists rigorously connected emotion to its physical underpinnings. Notable books included Sir Charles Bell's *The Anatomy and Philosophy of Expression* (1824) and Charles Darwin's *The Expression of the Emotions in Man and Animals* (1872) —both of which held that emotions are physiological and universal in nature. William James' seminal 1884 article "What is an Emotion?" posited that some emotions have "distinct bodily expressions"—affects—that can be categorized, measured, and analyzed.

Mid-20th-century researchers largely discarded such theories of universal, measurable emotions, in favor of those that held emotions to be learned and culturally determined. They enjoyed a revival with the work of Paul Ekman, whose 1978 publication of the "Facial Action Coding System" (FACS) provided a basis for deconstructing affect in facial expressions.

It was Rosalind Picard's 1995 paper "Affective Computing" that first fully asserted the value of affect in computing systems. An unusually philosophical paper, Picard said it was "roundly rejected" when submitted to a journal for publication. (One reviewer wrote, "this is the kind of stuff that fits in an



A screenshot showing video processing by the Computer Expression Recognition Toolbox (CERT).

IMAGE COURTESY OF JOSEPH F. GRAFSGAARD, NORTH CAROLINA STATE UNIVERSITY

in-flight magazine.") Picard, now director of the Affective Computing Research Group at the Massachusetts Institute of Technology (MIT), turned the paper into a book of the same name, found a publisher, and submitted the book with her tenure papers. "Back in '95, there was a lot of conjecture," she said. "We did not have real-time computer vision or vocal analysis: it was before cameras were ubiquitous. But I saw that, if computer agents could see when you are interested, or bored, or confused, that would lead to more intelligent interaction."

First, however, computers would need to learn how to "read" people, and FACS was only part of the puzzle.

Listening and Watching

Emotions spur responses throughout one's entire nervous system, not just in the face. Affective computing researchers have also studied effects on the voice and skin: each returns its own data complex, suitable for differing applications.

Aside from her FACS-based work, Picard has also trained computers to judge emotion based on non-visual signals. She became intrigued by statements from autistic people who said they felt increasing stress before a meltdown, but were frustrated by their inability to express it. The company Picard co-founded, Affectiva, developed a "Q Sensor" bracelet that measures electrodermal activity, and "Q Live" software to chart the results, thereby giving subjects a new way to express their interior lives.

One company attempting to plumb audio data is Tel Aviv, Israel-based Beyond Verbal, which has received two patents for what it calls "Emotions Analytics." Chief Science Officer Yoram Levanon believes the company's automated voice analysis can extrapolate emotional content in speech to help people make better-informed decisions. "Most decisions are made in between one- and three-tenths of a second," he said, "but cognition only begins after half a second. If I talk to you at 120 words per minute, you cannot process it cognitively—you must find another way to focus that." Beyond Verbal's products try to provide that focus by displaying text that comments on the speaker's emotions.

Facial expression recognition has captured the attention of more affective computing researchers. That is partly due to a wealth of developer toolkits that track the facial "Action Units" (AUs) described by FACS and its successors. These tools typically rely on one of two strategies: training a statistical shape model that aligns to characteristic curves of the face (such as eyes, mouth, and nose), as in an Active Appearance Model (AAM) or Constrained Local Model (CLM); or computing low-level vision features such as Gabor filters or Local Binary Patterns (LBP), to train machine-learned models of facial movements.

The Computer Expression Recognition Toolbox (CERT) was developed by the Machine Perception Laboratory at the University of California, San Diego, in collaboration with Ekman. According to Joseph Grafsgaard, a Ph.D. student in the Department of Computer Science at North Carolina State University, "An innovative aspect of CERT is that it is not only identifying facial landmarks, such as the shape of the mouth or eyes. It uses Gabor filters at multiple orientations, which are then mapped to facial action units via machine learning. This allows it to identify fine-grained facial deformations detailed in FACS, such as wrinkling of the forehead, or creasing around the mouth, eyes, or nose."

Another toolkit is produced by Emotient, which was founded by the original developers of CERT and includes Ekman on its board. According to co-founder and lead researcher Javier Movellan, the company's products are unusual in that they rely heavily on neural networks to "learn" different expressions, rather than on watching specific points on the face. He said, "People ask us, 'are you looking for wrinkles?' and we say, 'we don't know!' What we look for is a lot of data so the system can figure it out. Sometimes it takes us a while to figure out how the system is solving the problem."

So Tell Me: How *Do* You Feel?

Affective systems like these benefit greatly from human confirmation. Beyond Verbal "crowdsources" some of that research through its website by asking visitors to rate the results of their demos; Affectiva's facial action

ACM Member News

MULLER TAPS IDEAS OF THE RANK AND FILE VIA CROWDFUNDING

Two decades as an internationally recognized expert in participatory design and analysis have led IBM Master Inventor Michael Muller to conclude, "rank and file workers know things that aren't necessarily apparent to upper management."

Muller, who heads the Invention Development Team for the Collaborative User Experience Group in IBM Research and the IBM Center for Social Software in Cambridge, MA, used that conclusion to initiate a trial "enterprise crowdfunding project" for his group in 2012, to spur innovation and foster collaboration.

After building a website similar to RocketHub and Kickstarter, Muller put out a call for proposals, which were posted on the site. The Collaborative User Experience Group gave each person in Muller's group $100 to invest in the project of their choice (they weren't allowed to vote for their own ideas). "We expected 10-15% of the workforce to participate; instead, we received a 46% response rate; 500 people within IBM pledged money," he said.

The winning proposal was to have remote workers join Cambridge-based phone and videoconferences that they could not previously participate in, by allowing them to remotely access robots to participate virtually.

IBMers responded enthusiastically and three other groups have since initiated their own enterprise crowdfunding projects, raising from $31,000 to $150,000.

"The enterprise crowdfunding project helps forge new relationships. We're measuring success not in monetary savings, but in organizational outcomes and innovation," Muller said.

—Laura DiDio

coding experts compare their judgments to the system's. The benefit of crowdsourcing comes mostly during development, and to tune the system's algorithms. A different type of system starts with user queries, then correlates those self-stated moods with environmental measurements to show personal trends.

Emotion Sense started out as a voice-sampling application; speak into it, and it would attempt to determine your emotions. According to the project's Cecilia Mascolo, reader in Mobile Systems at the Computer Laboratory of the University of Cambridge, "After four years of that, we decided to use the human as a sensor for emotions, with something called 'experience sampling;' we ask the user's experience through questionnaires." Now available as an Android app, Emotion Sense matches mood to time of day, location, call patterns, and several other factors. So, for example, you might discover that your mood is generally better in the morning, or while in a certain location.

Mascolo says the ultimate goal is to "find the psychological markers out of all this data we are collecting, so we do not have to ask questions anymore. Maybe we can help psychologists and social scientists better understand what leads to being very happy, what leads to being very sad."

A similar project is Mood Sense (formerly MoodScope), a research collaboration between Microsoft Research Asia and Rice University. Like Emotion Sense, the prototype API queries user feelings, but correlates them only to communication behaviors (text, email, phone calls) and user interactions with applications and the Web browser. That is enough, according to Nicholas Lane, lead researcher at Microsoft Research Asia; "We find that how you use your phone, and what it means about your behavior in general, can be a strong signal of your mood or emotive state. That is the key innovation here." One benefit gained by narrowing the number of sensors monitored: Mood Sense is event-driven, and therefore consumes very little power. Says Lane, "The power savings are enormous over using the camera or audio, so if I wanted to run a psychology experiment, battery life on the subjects' phones wouldn't change from 40 hours to 10 hours."

"Affective computing was inspired by a desire to show more respect for people's feelings."

To Fear and Love the Ones Who Know You

While smartphone power loss is an inconvenience, those in affective computing point to a greater danger to the field: privacy violations from such technologies being used in a surreptitious or unethical way. Speaking about multi-sensor tracking ("sensor fusion") in the service of affective computing, Freescale Semiconductor Executive Director of Strategy Kaivan Karimi said, "The thing that can promote [sensor fusion] is all the cool stuff that comes along with it. The thing that can kill it is if information starts leaking and getting used the wrong way because there are no boundary conditions set."

Yet researchers seem to agree that some third-party disclosure would be appropriate. Microsoft Research senior research designer Asti Roseway said, "One size does not fit all. You could take certain populations, such as kids with autism or ADHD, or relationships such as teacher to student or parent to child, where having external notification of emotions is beneficial. On the other hand, you could be in a meeting where you get a private poke if you start feeling stressed, and nobody has to know. We have to be really smart about the context, but that takes trial and error."

Affective computing is likely to face many trials. Speaking about his company's face-reading technology, Emotient's Movellan said, "Interest is coming from various places, and some of it is surprising to us. Marketing people want to know if someone *really* likes a product. Educators want to know what parts of a video lecture people like, and where they are confused or not paying attention. The entertainment industry is also very interested; for example, to build personal robots that respond to your affect. Carmakers could put a camera in front of the driver to tell if you are looking dangerously fatigued and are about to crash. And health care providers can monitor facial expressions and alert caregivers when you have a depressed affect."

Picard acknowledged the potential for privacy abuses. "We have turned down work where people want to read emotions 'from a distance,' without them knowing," she said, "but affective computing was inspired by a desire to show more respect for people's feelings. So it would be incredibly hypocritical to use it in a way that disrespects people's feelings." C

Further Reading

Picard, R.W.
Affective computing. M.I.T. Media Laboratory, Perceptual Computing Section Technical Report No. 321 (1995).

Lathia, N., Rachuri, K.K., Mascolo, C., and Roussos, G.
Open source smartphone libraries for computational social science. In 2nd ACM Workshop on Mobile Systems for Computational Social Science (Zurich, Switzerland, 2013)

Wu, T., Butko, N.J., Ruvolo, P., Whitehill, J., Bartlett, M.S., and Movellan, J.R.
Multi-layer architectures for facial action unit recognition. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 42, no. 4, pp. 1027–1038 (2012)

Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., and Bartlett, M.
The computer expression recognition toolbox (CERT). In Proceedings of the 9th IEEE Conference on Automatic Face & Gesture Recognition, pp. 298-305 (Santa Barbara, California, 2011)

LiKamWa, R., Liu, Y., Lane, N. D., and Zhong, L.
MoodScope: Building a mood sensor from smartphone usage patterns. Proceedings of the ACM International Conference on, Mobile Systems, Applications, and Services (MobiSys) (Taipei, Taiwan, 2013)

Karimi, K.
The role of sensor fusion and remote emotive computing (REC) in the Internet of Things. White Paper (document SENFEIOTLFWP), Freescale Semiconductor, Inc. (June, 2013)

Tom Geller is an Oberlin, Ohio-based science, technology, and business writer.

Society | DOI:10.1145/2555808 Paul Hyman

'Peace Technologies' Enable Eyewitness Reporting When Disasters Strike

Ushahidi—or "testimony" in Swahili—has played a central role in coordinating responses to crises around the globe.

When Kenyans went to the polls in late 2007, the voting results were disputed, and ethnically aligned gangs slaughtered more than 1,100 people during weeks of violent unrest. A national media blackout followed.

Within days, an ad hoc group of Kenyan bloggers and software developers created the first instantiation of the Ushahidi ("testimony" in Swahili) platform, enabling people to share instant reports of violent incidents—via video, audio, photos, texts, SMS, emails, Twitter, and Web forms—and to visualize them on a real-time online map, to keep communities informed of the danger.

Just two years later, when a catastrophic magnitude 7.0 earthquake struck Haiti in 2010, Ushahidi sprung to prominence as a key player in what has become a growing field of information and communication technologies (ICTs)—so-called "peace technologies."

Deciding to put its real-time maps to good use once again, Ushahidi had a local Haitian radio station broadcast a free telephone number that forwarded texts to a group of volunteer graduate students working in a basement at Tufts University's Fletcher School in Boston.

By the time the Red Cross and others arrived on the scene in Haiti, an Ushahidi crisis map was available, online, and ready to guide rescue workers to red-dot locations from where there had come cries for help. In fact, U.S. Federal Emergency Management Agency (FEMA) chief Craig Fugate tweeted that the Ushahidi map of Haiti "was the most comprehensive and up-to-date map available to the humanitarian community." Then-U.S. Secretary of State Hillary Clinton, in a 2010 speech on Internet freedom, credited the map with leading a U.S. team to a seven-year-old girl and two women who had been buried under the rubble of a collapsed Haitian supermarket.

In one of its earliest projects, Ushahidi documented the demolition of settlements in Zimbabwe that the government claimed were illegal; human rights groups felt the demolitions were politically motivated. The before (above) and after photos show how the settlement in Porta Farm, Zimbabwe, was completely erased.

The Ushahidi platform itself has been described as a mash-up of SMS, email, and Google Earth, plus OpenStreetMap. To make sense of all the information that is being accumulated, a platform called SwiftRiver was later developed to enable the filtering and verification of real-time data.

In a mere five years, the Ushahidi platform has been deployed in more than 159 countries, and has been translated into more than 35 languages. Despite its roots in election monitoring, Ushahidi's software has been used for crisis response in Haiti, Pakistan, Chile, Indonesia, the Czech Republic, the U.S., and elsewhere, and for civil society actions having to do with the environment, harassment, and anti-corruption.

"We have seen a shift by governments and international bodies toward actively asking citizens to report incidents of violence in potential flashpoint situations, such as elections," observes Catherine Dempsey, a Ph.D. candidate at King's College London's Department of War Studies, who currently is researching community-led initiatives to prevent conflict using ICTs.

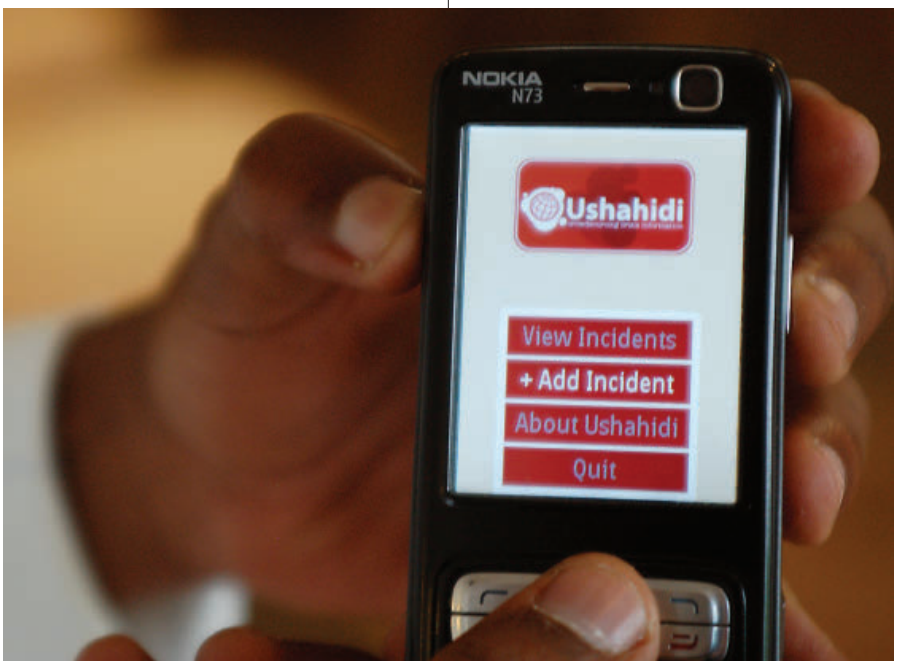
During the run-up to a constitutional referendum in 2010, the Kenyan government's National Steering Committee on Conflict Management and Peacebuilding asked citizens to report to an Ushahidi-inspired social media violence monitoring platform—the Uwiano Platform for Peace.

Ushahidi also set up an extensive election monitoring and violence monitoring project—called Uchaguzi—for the presidential elections in Kenya last March. The organization brought on board and trained more than 200 volunteers to work on the project, and partnered with state institutions and law enforcement as responders to emergency reports.

The use of Ushahidi has become so pervasive that a wiki now tracks "deployments of the week."

In fact, the use of Ushahidi has become so pervasive that a wiki now tracks "deployments of the week," which range in diversity from mapping Hurricane Sandy to keeping count of the world's mangroves to following the Palestinian crisis.

"Ushahidi is free and open-source software used for interactive mapping, data visualization, and information collection," explains Heather Leson, director of community engagement at Ushahidi, now a non-profit tech company. "Anyone can download the software or set up a map project. Maps have always been storytelling devices, but Ushahidi provides our users the opportunity to build the community network, customize, add layers of data, and even import/export content. With our code on GitHub, our community has created plugins, or simply forked the code to create their own projects."



An Ushahidi smartphone app, which allows users to report incidents, as well as view incident reports.

PHOTOGRAPH BY ERIK HERSMAN

Funding for Ushahidi's efforts comes from a potpourri of organizations; these include the Ford Foundation, the United States Institute of Peace (USIP) Center of Innovation for Science and Technology, the Omidyar Network, the Knight Foundation, the MacArthur Foundation, Google, the Mozilla Foundation, the Cisco Foundation, and others.

By enabling anyone to share geolocated information in real time, Ushahidi provides situational awareness for local communities, says Dempsey. She cites several examples of projects that are using Ushahidi technology for conflict prevention:

- SyriaTracker. Maps reports of violence in Syria's civil war, serving as one of the most comprehensive sources of information on the conflict.
- LERN. Liberia's Early Warning and Response Network maps incidents of violence across Liberia.
- Women Under Siege Syria. A Women's Media Center project to map incidents of sexualized violence in Syria.
- Harassmap. Uses Ushahidi to map and challenge the social acceptability of sexual harassment in Egypt.

While Ushahidi is probably the best-known peace technology, says Dempsey, there are quite a few other groups and organizations working in this field. These include:

- FrontlineSMS—A free and open-source text-messaging software to connect communities using SMS.
- Humanitarian OpenStreetMap Team (H.O.T.)—A project to connect the OpenStreetMap community with traditional humanitarian responders.
- CrisisTracker—A software platform that extracts situational awareness from torrents of public tweets during humanitarian disasters, combining automated processing with crowdsourcing to detect events and collate evidence.

▸ Standby Task Force (SBTF)—A volunteer community providing dedicated crowdsourcing, mapping, data scrambling, and technology testing support to local and international humanitarian and human rights organizations.

"Conflict prevention and peacebuilding work is so often about developing communication and information awareness," says Dempsey, "so new communication tools of all kinds are increasingly being used for these purposes."

She cites a group of civilians in Kyrgyzstan who carried out real-time fact-checking through a Skype chat group to debunk dangerous rumors that were fueling violent conflict in June 2010. In addition, an organization called Peace Provocateurs used Facebook and Twitter to correct mainstream media reports on clashes between Muslims and Christians in Indonesia in September 2011, helping to defuse tensions between those religious groups.

"Using technology for conflict prevention is nothing new," says Sheldon Himelfarb, director of the USIP's Center of Innovation for Science and Technology, which provided funding to Ushahidi during the Haitian crisis. "But what is different today—and this is the essence of Ushahidi—is that suddenly, for the first time in human history, everyone has the ability, no matter how poor you are, to send texts, data, photos, and words around the world with the push of a button. It is that phenomenon that Ushahidi captures by crowdsourcing information and allowing it to be put to use in communities who want to help themselves."

Ultimately, what is unique about Ushahidi, Himelfarb says, is that no one controls it.

"Ushahidi is a small, not-for-profit tech company that had one of the best ideas of the 21st century for capturing information," says Himelfarb, "and now, because it is open source and you can just go online and create an instance, people everywhere are using it. Ushahidi—or some version of it—is now being used in every single disaster around. If there is a tornado in Oklahoma, for example, I promise you there is a Ushahidi map up in about three minutes.

"Ushahidi has ushered in a whole new era of crowdsourced disaster response."

However, not all peace technologies focus on mapping using data collected on the ground; for instance, the American Association for the Advancement of Science (AAAS) Geospatial Technologies and Human Rights Project (GTHR) depends upon the gathering and analysis of high-resolution satellite imagery instead.

"We form partnerships with human rights organizations—like Amnesty International—and use the imagery to document areas of concern, often in places that are too dangerous to send people on the ground," says Susan Wolfinbarger, director of GTHR, which is part of the Scientific Responsibility, Human Rights, and Law Program of AAAS, an organization that also publishes the journal *Science*.

The GTHR recently released its report on Aleppo, Syria, where it used satellite imagery to document craters, destroyed buildings, destruction of cultural artifacts, and numerous roadblocks. Other GTHR projects include the documentation of the destruction of villages in Darfur, Sudan, and grave sites and the bombing of civilian shelters after the intense fighting in Sri Lanka in 2009.

"Ours is just one of the many ways that peace technologies are entering into the work of human rights groups," Wolfinbarger says, "the others being early warning systems, databases, the use of disruptive technologies for communications, disaster response, Internet privacy mechanisms, and so on."

Ushahidi, too, is working on new technologies, including BRCK, its first foray into hardware. Earlier this year, the organization launched a Kickstarter campaign to build a backup generator for the Internet. The small, lightweight box that is about the size of an actual brick works like a mobile phone, seamlessly switching between Ethernet, Wi-Fi bridging, and a 3G/4G connection whenever the user's preferred network is down. Plug in a SIM card and, with its eight-hour battery, the BRCK prototype provides a fully functioning network anywhere in reach of cellular service. Ushahidi's developers are currently searching for a manufacturer for BRCK.

Going forward, Ushahidi's goal is "to create free and open-source software to democratize information," says director of community engagement Leson.

"We want to help connect citizen voices to action," she says. "Ushahidi is now five years young. Our community and their reports are some of our biggest opportunities. We are diving into data to understand curation, signal-to-noise, and ways to use technology to assist this journey." C

Further Reading

Rahul Chandran and Andrew Thow
"Humanitarianism in the Network Age," 2013, the United Nations Office for the Coordination of Humanitarian Affairs (OCHA), https://docs.unocha.org/sites/dms/Documents/WEB%20Humanitarianism%20in%20the%20Network%20Age%20vF%20single.pdf

Francesco Mancini (ed.)
"New Technology and the Prevention of Violence and Conflict," April 25, 2013, the International Peace Institute, http://www.ipinst.org/media/pdf/publications/ipi_epub_new_technology_final.pdf

Heather Leson (ed.)
"Uchaguzi – Kenyan Elections 2013," April 24, 2013, Ushahidi, https://wiki.ushahidi.com/display/WIKI/Uchaguzi+-+Kenyan+Elections+2013

"Mapping Syria," a video posted July 16, 2013 by Ushahidi, at http://www.youtube.com/watch?v=NRG0PqYisZ8

Jessica Heinzelman and Carol Waters
"Crowdsourcing Crisis Information in Disaster-Affected Haiti," September 29, 2010, the United States Institute of Peace's Center of Innovation for Science, Technology, and Peacebuilding, http://www.usip.org/sites/default/files/SR252%20-%20Crowdsourcing%20Crisis%20Information%20in%20Disaster-Affected%20Haiti.pdf

Jessica Heinzelman, D. Roz Sewell, Jen Ziemke, and Patrick Meier
"Lessons from Haiti and Beyond: Report from the 2010 International Conference on Crisis Mapping," March 7, 2011, the United States institute of Peace, http://www.usip.org/sites/default/files/PB83.pdf

Jakob Rogstadius
"Introduction to CrisisTracker," July 7, 2012, http://vimeo.com/45366518

Paul Hyman is a science and technology writer based in Great Neck, NY.

DOI:10.1145/2542502

Michael A. Cusumano

Technology Strategy and Management

The Legacy of Steve Ballmer

Assessing the positive and negative components of the second Microsoft CEO's tenure.

WHERE DO YOU go after your company has achieved a 95% market share and a stock price at an historical peak? Add to this the departure of a visionary founder (see "The Legacy of Bill Gates," *Communications*, January 2009). Bill Gates stepped down as Microsoft's CEO in 2000, just when critical changes were about to disrupt Microsoft's economics: continuing growth of the Internet as a new applications and communications platform, handheld computers and cellphones starting to converge, growing popularity of free open-source software, and the beginning of software as a service and cloud computing. The most likely place to go is down, and that is what happened to Microsoft after Steve Ballmer took over as CEO.

The Gates-Ballmer relationship began when they were undergraduates at Harvard University during the mid-1970s. Ballmer graduated

On Aug. 23, 2013, it was announced that Steve Ballmer will step down as Microsoft CEO within 12 months.

PHOTOGRAPH BY MARIO TAMA/GETTY IMAGES

and then went to work for Proctor & Gamble, before attending Stanford's MBA program. He left after a year to join Microsoft in 1981 and help his old college dorm-mate grow the business. Ballmer went on to become enormously successful (and wealthy) himself. He was Gates' most trusted lieutenant for the next two decades, exhibiting great skills in marketing products, motivating the sales team, and building relationships with enterprise customers. The announcement in August 2013 that Ballmer will step down as CEO within 12 months makes this a good time to reflect on his legacy and how he has positioned Microsoft for the future.

Ballmer also announced a major acquisition to take place in 2014: Nokia's cellphone hardware and services business. Nokia's CEO for the past three years, Steven Elop, worked at Microsoft from 2008–2010 after being COO of Juniper Networks. He will rejoin the company as Executive Vice President of Microsoft's Devices and Services business unit. That puts Elop in a strong position to succeed Ballmer or at least become very influential within Microsoft—if the merger proceeds smoothly.

Positives

On the positive side, Ballmer did well on several key dimensions. First, and most importantly, Ballmer proved to be an able steward of the Windows platform—the "mother ship." This franchise is supported by enormous revenues from Windows, Office, and Windows server and tools. In fiscal year 2013, Microsoft had revenues of nearly $78 billion and operating profits of $27 billion; about 24% of sales came from Windows, 32% from Office and other business products, and 26% from Windows server and tools.[3] These three sectors also generated nearly all of Microsoft's operating profits. Historically, about 70% of Windows and Office sales also have been to enterprises (large corporate customers or personal computer manufacturers like Dell and Hewlett-Packard), not individuals. This means Microsoft is largely shielded from the volatile consumer market.

Second, Ballmer presided over the establishment of a new platform for home entertainment—the Xbox, launched in 2001, and currently the most popular video-game console. This business generated $10 billion in sales for Microsoft in 2013, though only $850 million in profit. Development started while Gates was still CEO in the late 1990s. However, Ballmer and his executive team, especially Robbie Bach (who left in 2010) and Don Mattrick, deserve lots of credit. The Xbox operating system is derived from Windows NT, but cannot run Windows programs and is optimized for graphics and video games. For Microsoft to create a new operating system incompatible with Windows was an important step to diversify and show it could build products aimed at being best in class, not just best at running Windows.

Ballmer proved to be an able steward of the Windows platform—the "mother ship."

Third, Ballmer presided over the extension of Windows and Office, as well as other packaged software products, to "the cloud," with Windows Azure, introduced in 2010, as well as the SkyDrive cloud hosting service, first offered as Windows Live in 2007. Microsoft's cloud-based infrastructure and development platform will become more important as enterprises increasingly access their software via the Web and rely on subscription pricing or build new hosted applications. Azure positions Microsoft reasonably well to maintain its enterprise accounts. However, young Web-based companies generally do not use Microsoft products and services because they are too expensive. They prefer Linux and other free open source products, or hosting via Amazon, Google, and others. This preference does not bode well for Microsoft. The startup firms of today will become the larger customers of the future and, in general, they are not Microsoft customers.

Fourth, Ballmer added some new platforms, brands, and complementary products and services. Most notable is Skype, for video communications, which Microsoft bought in 2011 for $8.5 billion. Under Ballmer, Skype has gone from a money-loser to generating some $2 billion in revenues last year, with about 300 million users. The sales and user base, which Microsoft could leverage for other product sales, helps justify the high acquisition price. Ballmer also bought Yammer in 2012 for $1.2 billion, giving Microsoft a presence in social networking for the enterprise. Overall, these acquisitions, along with the Nokia phone business, further diversify the company and should become more valuable. The Nokia acquisition will expose Microsoft more to the commodity consumer market. Nonetheless, since sales of smartphones and tablets now far outpace those of personal computers, a greater presence in mobile devices is necessary to grow revenues, albeit at the expense of profits.

Negatives

On the negative side, as CEO, Ballmer struggled or failed in several key areas. First, Ballmer was unable to rein in the many warring factions that fragmented the Windows group after Gates handed over the CEO job. With no one in charge to focus the huge teams (as many as 7,000 programmers and test engineers worked on Windows Vista), the operating system devolved into a massive pile of "spaghetti" code. There were too many bosses and too many bugs. It took leadership from the better-managed Office group to ship the poorly received Windows Vista, released in 2007 (after being originally scheduled for 2003), and then to right the ship with Windows 7, released in 2009 (see "What Road Ahead for Microsoft and Windows," *Communications*, July 2006, and "What Road Ahead for Microsoft the Company," *Communications*, July 2007). Steven Sinofsky rose to president of the Windows division, but he left after shipping Windows 8 in 2012, again making it unclear who was in charge of Windows.[1] In July 2013, Ballmer announced a reorganization, consolidating eight separate

product groups around software platforms, software applications, hardware devices, and online services. All major operating systems—desktop, server, and mobile—now report to one executive, Terry Myerson.[6] This move should improve coordination within Microsoft, but it will still be very hard to catch Apple and Google in smartphones, tablets, and Internet services.

Second, Ballmer did not sufficiently embrace broader changes in the technology landscape as he kept the company closely tied to Windows and PCs. This is another legacy inherited from Gates. Since Ballmer was not a software programmer, perhaps he should have been the one to make the technical and emotional break. A dozen years ago, Ballmer could have presented Microsoft as a platform company, not just a Windows company, and built other operating systems with workable bridges to Windows applications. This would have leveraged the Windows franchise but better enabled the company to move into tablets and smartphones.

Microsoft has made some efforts to evolve, but not aggressively, except for the Xbox. For example, Office has become a productivity platform for a billion users, but we do not see this application suite on smartphones and tablets running Apple's iOS or Google's Android. Office on non-Windows mobile devices is a huge gap in the market that Ballmer has not addressed. Perhaps Microsoft should have followed its own lead with Xbox and built a new operating system optimized for mobile devices, without worrying about Windows compatibility. But Ballmer found himself in an old Catch-22. Microsoft's main advantage over Apple and Google is the Windows platform. Smartphone and tablet customers who want to use their mobile devices as substitutes for PCs probably want to run standard Office and other Windows applications. At the same time, maintaining Windows compatibility has restricted Microsoft's ability to innovate and optimize in mobile software.

Third, while protecting the Windows franchise, Ballmer has managed to confuse users, hardware partners, application programmers, and industry analysts. In 2012, Microsoft introduced Windows RT, a reduced-instruction set operating system for tablets (and potentially smartphones) that runs on a long-battery-life ARM processor, like Google Android devices and the early Apple iPod and iPhone devices. RT tablets have met with poor sales and have little advantage over Google Android or iOS devices.[5] RT can only run new Windows 8 apps, and their number pales in comparison to the applications available for regular Windows or for Android and Apple devices. RT also requires a special emulation utility to run Windows 8 apps. Perhaps if we had seen Windows RT devices in 2006 or 2007, bundled with an RT version of Office, the new operating system might have had a chance. Now, smartphones running Google Android dominate sales, with about 80% of recent shipments.[2] Android tablet sales (more than 60% of recent shipments) have also passed the iPad (33%, down from over 75% in 2010).[4] In comparison, Microsoft's total presence in smartphones and tablets remains in the low single digits. Adding Nokia revenues will help, but they are now just a few percent of the market.

Ballmer remains an important figure because of what he did to help build the world's most successful software platform company.

Fourth, Ballmer let too much talent leave the company. The list of departures is very long, though it started while Gates was still CEO in the late 1990s. The list of new executive recruits under Ballmer is also short. One exception is Steven Elop, though he was unable to do much to reverse Nokia's decline after taking over there as CEO in 2010.

Finally, Ballmer never managed to make Microsoft's online business prosper, including MSN and the Bing search engine. This division also started under Gates in the mid-1990s, who then skillfully adapted the network to the Internet. MSN and Bing have provided Microsoft with lots of experience in how to run an online, ad-supported business. The division, nonetheless, has generated enormous red ink, including $12 billion in losses versus $8.7 billion in sales during the last three years. A sizeable part of these loses include charges taken against aQuantive, a $6 billion acquisition Ballmer made in 2007. This was an advertising software and services company that Microsoft failed to integrate, except for some of the ad software technology.

Conclusion

What to conclude? Ballmer leaves the CEO post with a mixed record. Microsoft's market value is about one-third less in 2013 than it was in 2000, despite tripling revenues and more than doubling profits under Ballmer. To be fair, growing revenues and profits while market value declined is common among high-tech companies that peaked at the height of the Internet boom, including Cisco and Intel. Nevertheless, Ballmer remains an important figure because of what he did to help build the world's most successful software platform company during the 1980s and 1990s. In other words, Ballmer's greatest legacy seems to be what he did *before* becoming Microsoft's CEO, not after.

References

1. Cusumano, M.A. What Sinofsky's departure suggests about the current state, and the likely future, of Microsoft. *MIT Technology Review* (Nov. 21, 2012); http://www.technologyreview.com/view/507746/what-sinofskys-departure-suggests-about-the-current-state-and-likely-future-of-microsoft/.
2. Etherington, D. Android nears 80% market share in global smartphone shipments, as iOS and BlackBerry sales slide, per IDC. TechCrunch (Aug. 7, 2013); http://techcrunch.com/2013/08/07/android-nears-80-market-share-in-global-smartphone-shipments-as-ios-and-blackberry-share-slides-per-idc/.
3. Microsoft Corporation. Form 10-K. United States Securities and Exchange Commission, Washington, D.C., June 30, 2013.
4. Pepitone, J. Apple's sliding mobile market share. CNN Money (Sept. 3, 2013); http://money.cnn.com/2013/09/03/technology/mobile/apple-market-share/.
5. Tibken, S. Top 10 biggest drawbacks of Windows RT. CNET (Oct. 23, 2012); http://news.cnet.com/8301-10805_3-57537730-75/top-10-biggest-drawbacks-of-windows-rt/.
6. Wingfield, N. Microsoft overhauls, the Apple way. *The New York Times* (July 11, 2013); http://www.nytimes.com/2013/07/12/technology/microsoft-revamps-structure-and-management.html?_r=0.

Michael A. Cusumano (cusumano@mit.edu) is a professor at the MIT Sloan School of Management and School of Engineering and author of *Staying Power: Six Enduring Principles for Managing Strategy and Innovation in an Unpredictable World* (Oxford University Press, 2010).

DOI:10.1145/2542503

Christopher S. Yoo

Law and Technology

Toward a Closer Integration of Law and Computer Science

Seeking better integration of the insights from the fields of law and technology.

The rate of technological change during the past few decades has been breathtaking. End users have adopted the Internet, smartphones, and tablets faster than any other consumer electronics product in history. The rapid diffusion of these technologies has transformed the way people work, shop, learn, play, and communicate.

Major technological changes inevitably have an impact on law. Just as the printing press revolutionized copyright and the telephone prompted new approaches to the Fourth Amendment, the digitization of all forms of content and the emergence of the Internet Protocol as the dominant platform for communication have led courts and legislatures to reexamine a wide range of legal issues.

During the Internet's early days, Silicon Valley spent little time worrying about whether the government would impose significant regulatory constraints. Initial disputes raised variations on familiar themes, such as whether the government can suppress online pornography, when an email message can constitute acceptance of a contract, and whether a company can be sued in a state simply because some of its residents purchased products from the company's website. More recent legal issues have increasingly arisen in increasingly complex technological contexts. Consider the following examples:

DNS and SOPA/PIPA. In late 2011 and early 2012, the U.S. Congress debated two legislative proposals known as the Stop Online Piracy Act (SOPA) and the Protect Intellectual Property Act (PIPA). Both bills would have required Internet service providers (ISPs) to use the Domain Name System (DNS) to block access to websites hosting content known to violate the copyright laws. At the House Judiciary Committee hearing on November 16, 2011, witness after witness repeatedly admitted they did not understand DNS well enough to discuss how the proposed legislation would interact with key technologies such as DNS Security Extensions

(DNSSEC), the well-established suite of protocols designed to help preserve the integrity of DNS by requiring that all answers to DNS requests be cryptographically signed.

Congestion management and network neutrality. Throughout the ongoing debates over network neutrality, both proponents and opponents agreed that any regulations should not prevent ISPs from taking reasonable steps to manage network congestion. Unfortunately, many of the people participating in the debate do not have a clear understanding of the way congestion is managed on the Internet. As a result, they fail to appreciate how the acknowledgment-based approach that has served as the foundation of congestion management since the late 1980s does not apply to increasing number of applications, such as VoIP and video, that rely primarily on the transport protocols known as the User Datagram Protocol (UDP), which does not use acknowledgments. The unfamiliarity with how congestion management works also obscures the fact that the central inference that Jacobson's algorithm (which presumes that a missing acknowledgment is a sign of congestion) is less appropriate for wireless networks. Such problems were relatively unimportant when communications consisted of email and Web browsing, which rely on the acknowledgment-based Transmission Control Protocol (TCP), and when most communications occurred over wireline technologies. In the modern era, however, video content and wireless transmission have become mission critical. Although solutions exist, such as the Datagram Congestion Control Protocol (DCCP) and hybrid Automatic Repeat reQuest (hybrid ARQ), these solutions require some important deviations from the existing architecture.

Location information. Many mobile devices (including all wireless devices connected to the public-switched telephone network) necessarily reveal information about end users' locations. At the same time, a growing number of applications are taking advantage of the geolocation Application Programming Interface (API) included in the latest version of HyperText Markup Language (HTML5), which discloses the end user's location. Location information can compromise an end user's security, as demonstrated by the advent of pleaserobme.com and other similar websites. In addition, some courts have held that the government can seize information made publicly available in this manner without obtaining a search warrant.

NSA's Project Bullrun. Edward Snowden claims that in addition to Project Prism, the U.S. National Security Agency (NSA) has been pursuing another program known as Project Bullrun designed to give it access to encrypted traffic. Reportedly, the NSA is inducing technology companies to insert vulnerabilities into commercial encryption systems, such as HyperText Transfer Protocol Secure (HTTPS), Secure Socket Layers (SSLs), and Virtual Private Networks (VPNs). Although Prism has gained far more notoriety, Snowden claims that Bullrun has been operating longer and has received significantly more funding. The integrity of encryption affects the scope of both federal privacy statutes and the Fourth Amendment, which turns largely on individuals' reasonable expectations of privacy.

The law has long struggled to keep pace with changes in technology. For example, more than 50 years passed after the invention of photography before the U.S. Supreme Court addressed whether photographs possessed sufficient originality to be copyrightable.[1] It took approximately the same amount of time for the Supreme Court to resolve the First Amendment status of cable television.[7] Some issues have never been fully resolved. Even though photocopying was successfully commercialized in the late 1940s, the Supreme Court still has yet to address how copyright applies to the technolgy, having deadlocked four-to-four in 1975 after one Justice recused himself.[9] The accelerating pace of technological change has made the complications associated with this lag all the more acute.

Just as technology has affected law, law has also affected technology. Legal restrictions have shaped and limited the ways innovations and business models can develop. The growing importance of legal considerations is perhaps best illustrated by the fact that increasing numbers of technology companies are establishing offices in Washington, D.C., to represent their interests before regulatory agencies and Congress. Prominent examples of how law affects technology and innovation include:

Data privacy. In contrast to the U.S. approach to privacy, which relies primarily on notice and consent, the European approach has been to place direct restrictions on the retention and uses of data. The hallmark of the European privacy regime has been to mandate that data can only be collected for limited purposes and for limited times. These restrictions place strict limits on the types of business models that companies with significant amounts of data can pursue and the types of innovative products that can emerge. In addition, many countries now require that their citizens' data remain within the country, which

forecloses a wide range of cloud computing solutions. Companies seeking to deploy new business models need to understand the precise boundaries of these restrictions.

Online video distribution. Copyright law has had a direct and dramatic effect on online video distribution. The Supreme Court's landmark 1984 *Sony* case established that end users could use videocassette recorders to make temporary copies of video content for later viewing.[8] Courts are beginning to consider whether copyright law is violated when the temporary copy is made by network providers instead of end users. This has a direct impact on technologies such as the Cablevision network digital video recorder,[2] Dish Network's Hopper, and new technologies for transmitting over-the-air broadcast television signals over the Internet, such as Aereo and Aereokiller.[4,10]

Patent policy. On June 4, 2013, a series of executive orders issued by President Obama threw a spotlight on the effect that patent policy can have on innovation. One area of controversy involves so-called non-performing entities (NPEs) that simply license and enforce patents without commercializing them directly. Another topic of ongoing debate concerns the mandate imposed by many standard-setting organizations (SSOs) that any patents included in a standard be licensed at fair, reasonable, and non-discriminatory (FRAND) rates. The SSOs, however, have provided precious little guidance as to the proper implementation of FRAND, and it was not until April 2013 that a federal court offered a comprehensive analysis of what FRAND means.[5] Other controversies surround the use of injunctions issued by courts and exclusion orders issued by the International Trade Commission.

Spectrum policy. The late Ronald Coase's 1959 article on the Federal Communications Commission represents a landmark in spectrum policy.[3] In essence, Coase recommended (1)using markets to allocate spectrum rights and (2) allowing individual rightsholders to redeploy spectrum to its highest and best use. While the first part of Coase's recommendations has become the prevailing orthodoxy, the second part of his recommendation remains unfulfilled. The vast majority of the spectrum remains encumbered by use restrictions that limit the technologies that can be deployed in any particular band. At the same time, an active debate exists over whether the federal government should set aside more spectrum available for unlicensed uses.

The next logical step would be to embed the interaction between law and policy deeper into the fabric of both fields.

The result is that law and engineering can no longer remain compartmentalized into separate spheres. A world in which innovation affects law and law in turn recursively affects innovation creates a need for decision makers and professionals who have a firm grounding in both spheres. The need for greater expertise does not arise only when dealing with the government: innovators and individuals also need to understand the interaction between law and technology when organizing their private affairs.

The problem is the connections between the two fields remain nascent and underdeveloped, often restricted to a few observations about policy implications offered in the introduction and conclusion of technical articles. Some organizations have begun to bridge the gap. For example, ACM's U.S. Public Policy Council (USACM) plays a critical role in providing government policymakers with information about issues relating to technology policy, although it remains predominantly an organization of engineers. In addition, the Electronic Frontier Foundation (EFF) brings together lawyers, policy analysts, and technologists to influence the law through litigation and white papers. EFF focuses exclusively on advocacy, and its engagement with technological considerations remains the exception and not the rule. The myriad academic centers that have sprung up to study law and technology focus primarily on law and have generated only weak ties to engineering schools.

The next logical step would be to embed the interaction between law and policy deeper into the fabric of both fields. For example, we could change the way we educate both engineers and lawyers. Rather than focusing primarily on one field and treating the other peripherally, programs could give students advanced training in both fields and could be designed in a way that requires students to grapple with both disciplines simultaneously. Indeed, noted Judge Richard Posner's most recent book calls for precisely this type of reform, pointing to the new program at the University of Pennsylvania as a pioneer in integrating technology into legal education.[6] Moreover, innovative interdisciplinary research needs conferences, journals, and other similar institutions to provide an intellectual home for the burgeoning field. Ultimately, faculty would emerge with advanced training in both disciplines, a vision that to date remains more dream than reality.

If successful, this movement will create a new generation of scholars and scholarship that will integrate the insights of both law and engineering in a pathbreaking and dynamic way. Such an approach is essential if our society is to continue to enjoy the benefits of economic and technological progress. C

References

1. *Burrow-Giles Lithographic Co. v. Sarony*, 111 U.S. 53 (1884).
2. *Cartoon Network LP v. CSC Holdings, Inc.*, 536 F.3d 131 (2d Cir. 2008).
3. Coase, R.H. The Federal Communications Commission. *Journal of Law and Economics 2* (1959), 1–40.
4. *Fox Broadcasting Co. v. Dish Network L.L.C.*, 723 F.3d 1067 (9th Cir. 2013).
5. *Microsoft Corp. v. Motorola, Inc.*, No. C10-1823JLR, 2013 WL 2111217 (W.D. Wash. Apr. 25, 2013).
6. Posner, J. *Reflections on Judging*. Harvard University Press, 2013, 347–348
7. *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417 (1984).
8. *United States v. Playboy Entertainment Group, Inc.*, 529 U.S. 803 (2000).
9. *Williams & Wilkins Co. v. United States*, 420 U.S. 376 (1975).
10. *WNET, Thirteen v. Aereo, Inc.*, 722 F.3d 500 (2d Cir. 2013) (Chin, J., dissenting from denial of rehearing en banc).

Christopher S. Yoo (csyoo@law.upenn.edu) is the John H. Chestnut Professor of Law, Communication, and Computer & Information Science and the founding director of the Center for Technology, Innovation and Competition at the University of Pennsylvania.

DOI:10.1145/2542504

Thomas Haigh

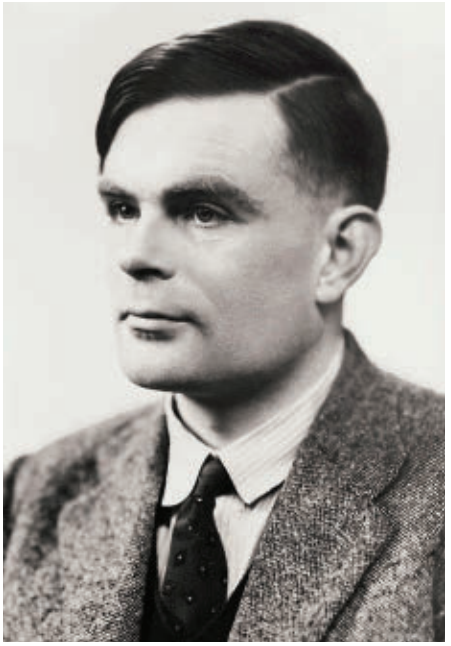
Historical Reflections

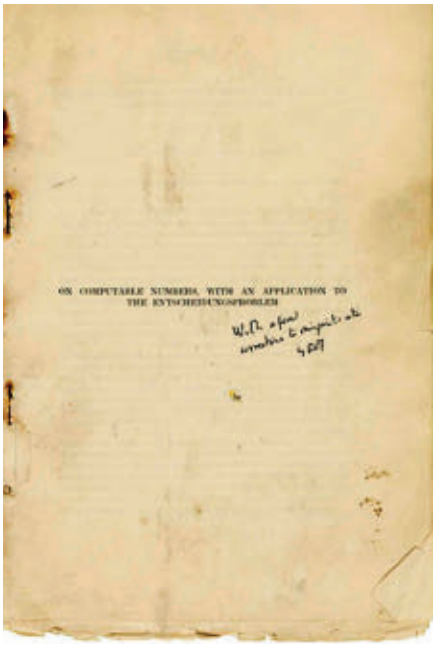
Actually, Turing Did Not Invent the Computer

Separating the origins of computer science and technology.

THE 100TH ANNIVERSARY of the birth of Alan Turing was celebrated in 2012. The computing community threw its biggest ever birthday party. Major events were organized around the world, including conferences or festivals in Princeton, Cambridge, Manchester, and Israel. There was a concert in Seattle and an opera in Finland. Dutch and French researchers built small Turing Machines out of Lego Mindstorms kits. Newspaper and magazine articles by the thousands brought Turing's life story to the public. ACM assembled 33 winners of its A.M. Turing Award to discuss Turing's ideas and their relationship to the future of computing. Various buildings, several roads, and at least one bridge have been named after him.

Dozens of books with Turing's name in the title were published or reissued. Turing was so ubiquitous that even George Dyson's book about John von Neumann was titled *Turing's Cathedral*, becoming the first book on the history of information technology to reach a broad audience since the one about Nazis with punched card machines. Publishers are well aware there is a strong audience for books about Nazis. The public's hunger for books about mathematicians and computer scientists is less acute, making Turing's newfound commercial clout both unlikely and heartening.





Alan Turing (left); the cover page of Turing's paper "On computable numbers, with an application to the Entscheidungsproblem" (right).

Still, as this flood of Turing-related material begins to recede it is time to clean up some of the rather bad smelling historical claims left in our metaphorical basement. Column space is short, so I will focus here on the idea that Turing invented the computer. Very short version: it is wrong.

In case you spent 2012 in a maximum-security prison or meditating in a Tibetan monastery, let me briefly summarize the computer-related high points of Turing's actual career. In 1936, just two years after completing his undergraduate degree, he introduced the concept now called the Turing Machine in a paper called "On computable numbers, with an application to the Entscheidungsproblem." This has since become the main abstract model of computation used by computer scientists. During the Second World War Turing made several vital contributions as part of the British team try-

TURING PHOTO COURTESY OF NATIONAL PORTRAIT GALLERY; PAPER IMAGE COURTESY OF CHRISTIE'S

ing to decipher intercepted German communications, which were encoded using specialized machines and had been thought unbreakable. Immediately after the war Turing designed an electronic computer, the ACE, for the National Physical Laboratory. A series of machines based on the design were eventually built, including one of the first commercial computer models, though Turing departed for the University of Manchester before serious construction began. He worked there with one of the earliest modern computers, but soon turned to more abstract and philosophical questions. Pondering the possibility of what we would now call artificial intelligence, Turing proposed we should judge a computer intelligent if someone could not reliably tell it from a real human after conducting a typed conversation with both. This procedure is now called the "Turing Test." Turing's career came to an abrupt end in 1954 with his death, usually attributed to suicide following various humiliations inflicted by the authorities after a legal conviction for homosexuality.

That is a remarkable career by any measure, with enough tragedy and genius to hook a broader audience and make Turing an unlikely gay icon. I do not have the expertise to evaluate the common claim that Turing's work shortened the war by several years but even a more cautious evaluation of the impact of his wartime accomplishments would make him a mistreated national hero. To celebrate Turing is therefore to celebrate freedom and decency, as well as genius. Let's just make sure we do our cheering in a historically responsible manner.

Retroactively Founding Computer Science

Turing provided a crucial part of the foundation of theoretical computer science. There was no such thing as computer science during the early 1950s. That is to say there were no departments of computer science, no journals, no textbooks, and no community of self-identified computer scientists. An increasing number of university faculty and staff were building their careers around computers, whether in teams creating one-off computers or in campus computer centers serving users from different scientific disciplines. However, these people had backgrounds and appointments in disciplines such as electrical engineering, mathematics, and physics. When they published articles, supervised dissertations, or sought grants they had to be fit within the priorities and cultures of established disciplines. The study of computing always had to be justified as a means, not as an end in itself.

I could fill many columns doing nothing more than skewering ridiculous things written about Turing, many of them by people who ought to know better.

Ambitious computer specialists were not all willing to make that compromise and sought to build a new discipline. It was eventually called computer science in the U.S., though other names were proposed and sometimes adopted. To win respectability in elite research universities the new discipline needed its own body of theory. The minutiae of electronic hardware remained the province of engineering. Applied mathematics and numerical analysis were tied too closely to the computer center tradition of service work in support of physicists and engineers. Thus, the new field needed a body of rigorous theory unique to computation and abstracted from engineering and applied mathematics.

Turing was not, in any literal sense, one of the builders of the new discipline. He was not involved with ACM or other early professional groups, did not found or edit any journal, and did not direct the dissertations of a large cohort of future computer scientists. He never built up a laboratory, set up a degree program, or won a major grant to develop research in the area. His name does not appear as the organizer of any of the early symposia for computing researchers, and by the time of his death his interests had already drifted away from the central concerns of the nascent discipline.

When building a house the foundation goes in first. The foundations of a new discipline are constructed rather later in the process. Turing's 1936 paper was excavated by others from the tradition of mathematical logic in which it was originally embedded and moved underneath the developing new field. In several papers historian Michael S. Mahoney sketched the process by which this body of theory was assembled, using pieces scavenged from formerly separate mathematical and scientific traditions. The creators of computer science drew on earlier work from mathematical logic, formal language theory, coding theory, electrical engineering, and various other fields. Techniques and results from different scientific fields, many of which had formerly been of purely intellectual interest, were now reinterpreted within the emerging framework of computer science.[a] Historians who have looked at Turing's influence on the development of computer science have shown the relevance of his work to actual computers was not widely understood in the 1940s.[1,4,5]

Turing's 1936 paper was one of the most important fragments assembled during the 1950s to build this new intellectual mosaic. While Turing himself did see the conceptual connection he did not make a concerted push to popularize this theoretical model to those interested in computers. However, the usefulness of his work as a model of computation was, by the end of the 1950s, widely appreciated within large parts of the emerging computer science community. Edgar Daylight has suggested that Turing's rise in prominence owed much to the embrace of his work by a small group of theorists, including Saul Gorn, John W. Carr, and Alan J. Perlis, who shared a particular interest in the theory of programming languages.[3] His intellectual prominence has been increasing ever since, a status both reflected in and reinforced by ACM's 1965 decision to name its premier award after him.

a See part three of Mahoney's *Histories of Computing*, cited in the Further Reading section at the end of this column.

So Who Did Invent the Computer?

This question, asked at a party, will cause any responsible historian of computing to blanch and mumble an excuse before scurrying to the safety of the drinks table. The whole way we write and think about the computers of the 1940s is an attempt to avoid having to provide a single answer to that question. Instead we award each early machine, and its main inventor(s), a metaphorical trophy engraved with a phrase such as "first general-purpose automatic electronic digital computer." These trophies adorn the figurative mantelpieces of John Atanasoff, Konrad Zuse, J. Presper Eckert, John Mauchly, Tom Kilburn, Tommy Flowers, Howard Aiken, and Maurice Wilkes. Those who focus on designs, rather than actual functioning machines, can and do make the case for Charles Babbage and John von Neumann. A colleague once joked to me that we should identify and honor the earliest computer never to be claimed as the first computer.

The story behind all those "firsts" goes like this. From the late 1930s to the mid-1940s, a number of automatic computing machines were built. Their inventors often worked in ignorance of each other. Some relied on electromechanical relays for their logic circuits, while others used vacuum tubes. Several machines executed sequences of instructions read one at a time from rolls of paper tape. Thanks in part to a series of legal battles around a patent granted on the ENIAC these machines dominated early discussion of the history of computing and their creation has been well documented.

The "modern" or "stored program" computers from which subsequent computers evolved were defined by two interrelated breakthroughs. On an engineering level, computer projects of the late 1940s succeeded or failed based primarily on their ability to get large, fast memories to work reliably. The first technology proposed, by Eckert who oversaw the engineering of ENIAC at the University of Pennsylvania, was the mercury delay line. Freddy Williams, working on the computer project at Manchester University, was the first to successfully store bits on a cathode ray tube. These were the two dominant high-speed memory technologies until the mid-1950s.

On a conceptual level, the breakthrough was inventing what we could now call a computer architecture able to take advantage of the flexibility of these new memories. Historians agree that the first wave of modern computers under construction around the world during the late 1940s were all inspired by a single conceptual design, an unpublished typescript cryptically titled "First Draft of a Report on the EDVAC." This unfinished document summarized discussions among the team working on a successor to ENIAC. Its title page named only John von Neumann as its author, though the extent to which he personally created the ideas within rather than summarizing the team's progress has been much debated. Turing produced his own ACE design only after reading and being influenced by this document, though his approach diverges in several interesting respects from von Neumann's.

Arguments For Turing

As historians followed this progression of machines and ideas they found few mentions of Turing's theoretical work in the documents produced during the 1940s by the small but growing community of computer creators. Turing is thus barely mentioned in the two main overview histories of computing published during the 1990s: *Computer* by Campbell-Kelly and Aspray, and *A History of Modern Computing* by Ceruzzi.

Much of the overstatement of Turing's role, in newspaper articles or by participants in online discussion, is based on simple misunderstandings. For example, a series of Colossus computers was used by the British for wartime codebreaking work. These were the first electronic digital computers to work properly. People often assume, incorrectly, that Turing must have designed Colossus because he worked at the same secret facility doing closely related work.

I could fill many columns doing nothing more than skewering ridiculous things written about Turing, many of them by people who ought to know better. We will learn more by looking at the best-supported, most careful arguments in favor of the idea that Turing invented the computer. The philosopher Jack Copeland has been one of the most passionate and industrious boosters of Turing's role in recent years, un-

leashing a book on Turing's ACE computer, another on Colossus, a collection of Turing's work, a website full of archival Turing documents, and a series of journal articles. His work continues the influential legacy of logician Martin Davis, whose history of computing *Engines of Logic* presented the universal Turing machine as the crucial advance behind the modern computer.

A painstaking and easily accessible summary of the case for Turing comes is "Alan Turing: Father of the Modern Computer" published by Copeland and Diane Proudfoot in an online journal edited by Copeland.[2] This claims that the "fundamental conception" embodied in the "First Draft Report" came from Turing, and that von Neumann himself "repeatedly emphasized" this. Copeland also believes that "right from the start" Turing was interested in building an actual computer based on the conceptual mechanism described in his 1936 paper. This extends a recent trend, seen for example in George Dyson's book, to write about the teams working to build computers in the late-1940s as if they launched their projects primarily to build practical realizations of Turing's abstract machine.

Copeland is deeply knowledgeable about computing in the 1940s, but as a philosopher approaches the topic from with a different perspective from most historians. While he provides footnotes to support these assertions they are often to interviews or other sources written many years after the events concerned. For example, the claim that Turing was interested in building an actual computer in 1936 is sourced not to any diary entry or letter from the 1930s but to the recollections of one of Turing's former lecturers made long after real computers had been built. Like a good legal brief, his advocacy is rooted in detailed evidence but pushes the reader in one very particular direction without drawing attention to other possible interpretations less favorable to the client's interests.

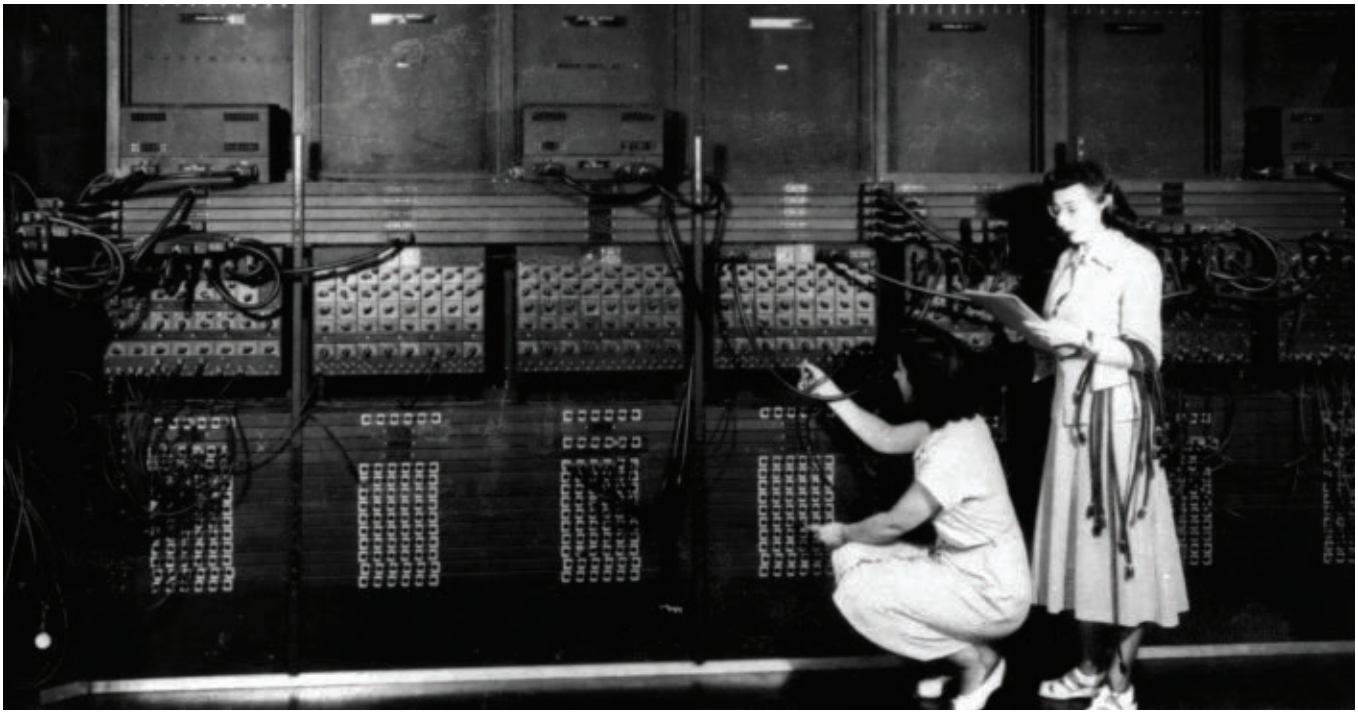
Theory vs. Practice

Arguments of this kind raise fundamental issues about the connection of theory and practice. Are abstract, theoretical insights more fundamental than pragmatic, engineering-based advances? Must theoretical breakthroughs precede and guide practical ones? For a computer scientist, in particular, it is easy to assume that Turing's theoretical work was as centrally important to the computer designers of the 1940s as it later becomes within computer science. There is also something undeniably attractive in the story of a lone genius who anticipates the rest of the world by many years.

Turing's work was not completely unknown in the 1940s. There is, for example, reliable evidence that von Neumann was aware of the now-famous paper and shared Turing's interest in the underlying mathematical questions it addressed.

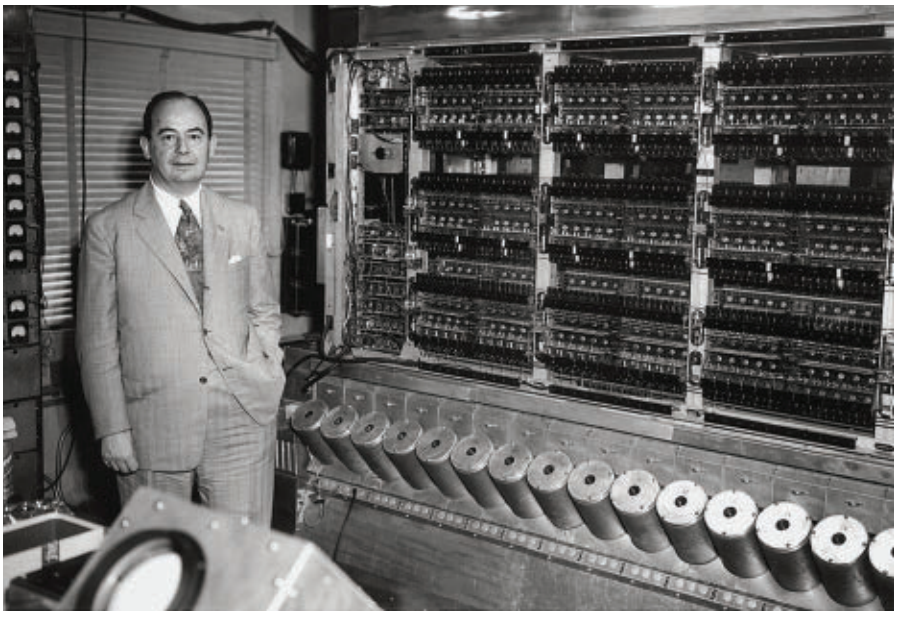
Where one might leap into fantasy is by asserting the cluster of ideas contained in von Neumann's 1945 "First Draft" are merely a restatement, or at most an elaboration, of Turing's earlier work on computability. Judge for yourself, by placing side by side Turing's 1936 "On Computable Numbers..." and "First Draft of a Report on the EDVAC." They are easy to find with Google, though you might want to pour yourself a fortifying beverage first as neither is particularly easy reading.

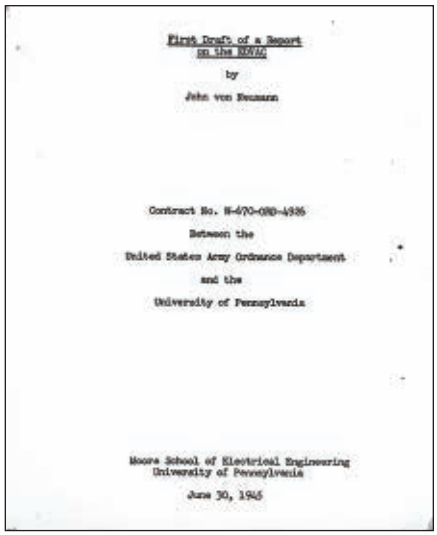
The former is a paper on mathematical logic. It describes a thought experiment, like Schrödinger's famous 1935 description of a trapped cat shifting between life and death in response to the behavior of a single atom. Schrödinger was not trying to advance the state of the art of feline euthanasia. Neither was Turing proposing the construction of a new kind of calculating machine. As the title of his paper suggested, Turing designed his ingenious imaginary machines to address a question about the fundamental limits of mathematical proof. They were structured for



U.S. ARMY PHOTO FROM THE ARCHIVES OF THE ARL TECHNICAL LIBRARY

Two programmers wiring the right side of the ENIAC with a new program.





John von Neumann with the IAS computer circa 1951 (left); cover page of von Neumann's "First Draft of a Report on the EDVAC" (right).

simplicity, and had little in common with the approaches taken by people designing actual machines.

Von Neumann's report said nothing explicitly about mathematical logic. It described the architecture of an actual planned computer and the technologies by which it could be realized, and was written to guide the team that had already won a contract to develop the EDVAC. Von Neumann does abstract away from details of the hardware, both to focus instead on what we would now call "architecture" and because the computer projects under way at the Moore School were still classified in 1945. His letters from that period are full of discussion of engineering details, such as sketches of particular vacuum tube models and their performance characteristics.

The phrase "stored program concept" has sometimes been used to encapsulate the content of the "First Draft" report, but this underplays its actual impact by implying it held just one big idea. In fact it provided a wealth of intertwined ideas and details. In my current work with Mark Priestley and Crispin Rope I have found it useful to separate these into three main areas.[b] The first, the "EDVAC Hardware Paradigm" described an all-electronic binary computer with a much larger memory than anything ever built previously. The second, the "von Neumann Architecture Paradigm," set out the basic structure of the modern computer: special-purpose registers on which all operations were performed and from which data was exchanged with main memory, separation of arithmetic functions from control functions from memory units, only one action performed at a time, and so on. The third, the "Modern Code Paradigm," concerns the nature and capabilities of its instructions. For example, instructions were expressed as through a small vocabulary of operation codes followed by argument or address fields. These were held in the same numbered memory cells as data. While executed by default in a particular sequence, the machine could jump out of sequence and the destination of this jump could be modified as the program ran based on the state of the computation.

Taken together, von Neumann's cluster of ideas guided the construction of computers that were much cheaper, smaller, more reliable, and more flexible than their predecessors. ENIAC, the first general-purpose electronic digital computer, used almost 18,000 vacuum tubes. The more tubes a machine held the more expensive it was to build and, as they eventually burn out, the less reliable. Its immediate successors held 1,000 or 2,000 tubes yet could handle problems of greater logical complexity and were easier to program. This efficiency made possible the construction of computers in cash-strapped Britain following the war, and made computers affordable and useful enough that they were rapidly turned into commercial products and applied to business tasks as well as scientific computations.

The universal Turing Machine has appealed to theorists from the 1950s onward.

According to Copeland, "the fundamental conception of the stored-program universal computer" was Turing's. Von Neumann merely "wrote the first paper explaining how to convert Turing's ideas into electronic form."[c] But what actually would have been different about von Neumann's "First Draft" report if Turing had never written his now famous paper? My answer to that question is: nothing (with the possible exception of the neuron notation he appropriated to describe logic gates, whose creators cited Turing).

Copeland has gone so far as to argue the basic idea of a single machine that could do different jobs when fed

b "Reconsidering the Stored Program Concept," forthcoming in *IEEE Annals of the History of Computing*.

c See http://www.huffingtonpost.com/jack-copeland/what-apple-and-microsoft-_b_3742114.html

PHOTOGRAPH COURTESY OF THE SHELBY WHITE AND LEON LEVY ARCHIVES CENTER, INSTITUTE FOR ADVANCED STUDY (IAS); PAPER IMAGE COURTESY OF THE UNIVERSITY OF PENNSYLVANIA ARCHIVES

different instructions can be traced to Turing. But Charles Babbage had that idea long before, and as mentioned earlier, several computers controlled by sequential instruction tapes had already been constructed with no influence from Turing and were well known to von Neumann before he wrote his report. EDVAC went far beyond this to store a program in addressable internal memory rather than on a sequential instruction tape. To suggest this advance came from Turing is odd, as the machine Turing described had no internal writable memory and took its instructions from a tape. Von Neumann brought a concern with logic and preference for minimal, general-purpose mechanisms to the design of EDVAC but he did not need Turing to teach him that. He was a mathematic genius with a deep pragmatic streak and an astonishing track record of productive collaborations across a huge range of fields.

Turing's 1936 paper lacks many novel and fundamental features found in the "First Draft" such as addressable memory locations. Neither did Turing describe instruction codes followed by arguments, the building blocks of computer programs. The suggestion that the EDVAC design was merely a conversion of Turing's paper implies these features are trivial, and the single important idea in each document is that code and data should be treated interchangeably so programs can modify themselves. Yet while Turing's paper showed one machine could, in modern terms, emulate the functioning of another it never described a machine altering its own instructions. Furthermore, at the very end of the "First Draft" von Neumann expressly forbade EDVAC from overwriting the operation fields in its instructions, even though he relied on modifications to their address fields to accomplish basic operations such as conditional branching. This address modification was a very influential idea in the "First Draft," but was, of course, absent from Turing's paper as his machines did not use addresses. In other words, the capability for unrestricted self-modifying code von Neumann is said to have copied from Turing is something Turing did not describe and von Neumann's design explicitly prohibited.

Computer Science vs. Computing

Our urge to believe the computer projects of the late 1940s were driven by a desire to implement universal Turing machines is part of a broader predisposition to see theoretical computer science driving computing as a whole. If Turing invented computer science, which is itself something of an oversimplification, then surely he must have invented the computer. The computer is, in this view, just a working through of the fundamental theoretical ideas represented by a universal Turing machine in that it is universal and stores data and instructions interchangeably.

This line of thinking blurs the fundamental distinction between building something and modeling it. Copeland shows that as early as 1949 von Neumann alluded to Turing's abstract model of computation as an interesting proof that automata with a certain "minimum level of complexity" could simulate each other's functioning. Yet finding an abstraction useful or provocative as a model of a particular real system does not imply the design of the real system was patterned on the abstraction. An abstraction, ultimately, is useful because of what it leaves out.

To focus on historical computers primarily as embodiments of logical ideas, ignoring the trade-offs their creators made when faced with limited resources and unproven technologies, is to abstract away from the information needed to understand their history and development. Progress in electronic engineering, particularly in memory technologies, created the circumstances in which it began to make sense to think about high-speed digital computers in which instructions were stored electronically. In turn, ideas about the best way to design these machines drove further progress in component technologies and engineering methods.

The universal Turing Machine has appealed to theorists from the 1950s onward precisely because it abstracts away from the complexity of real computer architectures and decouples questions of computability from those of design and engineering. This has been enormously useful for computing theorists, both technically and sociologically. Yet, paradoxically, the world seems increasingly eager to locate the origin of the computer in a mathematical abstraction adopted precisely because it hid all the messy issues of architecture and engineering needed to make any real computer function. Hardware and software are interchangeable to the theorist, but not to the historian. C

Further Reading

Aspray, W.
John von Neumann and the Origins of Modern Computing. **MIT Press, 1990. A thorough and careful survey of von Neumann's many contributions to early computing, including his work on the "First Draft of a Report on the EDVAC."**

Copeland, J.
Turing: Pioneer of the Information Age. **Oxford, 2013. A concise summary of Copeland's work on Turing's ideas and their legacy. He has produced related volumes on Turing's planned ACE computer and the wartime Colossus work.**

Hodges, A.
Alan Turing: The Enigma (Centenary Edition). **Princeton University Press, 2012. An updated edition of the monumental biography that originally put Turing on the road to broader fame.**

Lavington S., Ed.
Alan Turing and His Contemporaries: Building the World's First Computers. **British Informatics Society, 2012. A concise and clearly written expert history, honoring Turing's accomplishments and placing them in the context of British computer developments during the 1940s.**

Levy, P.
"The Invention of the Computer." In Serres, M. (Ed.) ***A History of Scientific Thought.*** **Blackwell, 1995. Concise and thoughtful in its summary of key early computers and their relationship to technologies, applications, and Turing.**

Mahoney, M.S., Ed. Haigh, T.
Histories of Computing. **Harvard University Press, 2011. Section three of this book, "The Structures of Computation," is a provocative selection of papers on the origins of theoretical computer science and its relationship to computation and simulation.**

References

1. Akera, A, *Calculating a Natural World*, MIT, 2006.
2. Copeland, B.J. and Proudfoot, D. Alan Turing: Father of the modern computer. *Rutherford Journal* 4, 2011–2012; http://www.rutherfordjournal.org/article040101.html.
3. Daylight, E.G. Towards a historical notion of 'Turing—The father of computer science.' To appear in *History and Philosophy of Logic*; www.dijkstrascry.com/TuringPaper.
4. Mounier-Kuhn, P. Logic and computing in France: A late convergence. *International Symposium on History and Philosophy of Programming*; http://www.computing-conference.ugent.be/file/12.
5. Priestley, M. *A Science of Operations.* Springer, 2010.

Thomas Haigh (thaigh@computer.org) is an associate professor of information studies at the University of Wisconsin, Milwaukee, and chair of the SIGCIS group for historians of computing.

DOI:10.1145/2542505

Phillip G. Armour

The Business of Software
Estimation Is Not Evil

Reconciling agile approaches and project estimates.

PHOTOGRAPH BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

ACCORDING TO RON JEFFRIES, estimation—as it is usually practiced—is "evil."[a] After making allowance for the hyperbole in the title (which does encourage people to read one's article) Jeffries makes the case that agile teams are often "*excessively concerned*" with estimating the work they need to do. More correctly, these are the "undistinguished" agile teams that may have achieved some improvement in performance due the adoption of agile, but have not fully achieved their potential, whatever that is. And estimating their work is getting in the way of this progress. But is this true?

The record of the business of software living up its promises, or promises made on its behalf, is generally considered to be poor. Software development has been accused of being too slow, too error-prone, and too costly. Though as Tom DeMarco observed in *Why Does Software Cost So Much?*[2] we might reasonably ask: compared to what? If software were truly too expensive wouldn't market forces have replaced it with something else?

Of course, everyone wants software development to be quicker, less expensive, and higher quality while requiring fewer people. Such expectations of improvement are quite reasonable. But expectations of accountability of software developers are also reasonable and, like it or not, companies will require it.

Some proponents of agile take the view that the approach is so radically different from other development methods that certain practices, including estimation, no longer apply or are pernicious. I think that underlying this view are some misunderstandings about the nature of software, estimation, and the approaches themselves.

Requirements vs. Estimation

The classic "Waterfall" life cycle approach tends to be an object of ridicule these days. This is usually coupled with distaste for *Big Process Up-Front* where *Process* is requirements, design or, well, process. This approach is considered old and it is. However, even in the ancient waterfall days, very few projects could or did legitimately predefine 100% of their requirements and design prior to building something. Projects that did not flex with changing requirements were not successful as much then as now. That said, the waterfall model was never really a way of working. It was not so much a development model as a *management* model that allowed a simplified basis on which to track projects. Work was rarely if ever finished on an unambiguous date, while most requirements might have been defined up front others were identified in later phases, and necessary reworking occurred throughout the life

a See R. Jeffries, "Estimation is Evil," *Pragmatic Programmer*; http://pragprog.com/magazines/2013-02/estimation-is-evil.

cycle all of which contradict the naive model assumptions.

While 100% defining requirements up front would be nice, it is not necessary either to build a system or to estimate it. Some agilists' negative reaction to the practice of estimation might be simply a misplaced allergy to fixed-requirements-up-front, as if both estimates and requirements act as constraints. Estimates are always uncertain; uncertainty is baked into the very definition of the word. Estimates are not values—they are ranges of uncertainty. The extent to which requirements are uncertain is about the same extent to which an estimate is uncertain. This is pretty intuitive: If we do not know what we want to do we cannot expect to figure out precisely how long it will take or what it will cost. But if we do not have a baseline of *some* expectations we will not even know where we are or where we are going. In addition, there is a disconnect between the business and the development activity.

Workflow vs. Resourcing

This is a major source of misunderstanding between agile practitioners and management. Much of the agile approach really concerns detailed workflow planning: how to decide what to do over the next few increments of time and how to generate the most value in that time. This is very good. But the resources allocated to the project to be used over these increments of time come from somewhere. Someone is in charge of them and is accountable to the company for them. And this person wants to know how many resources will be required and what the return will be. One of the jobs of estimation is to generate the information that will allow this calculation. Companies will not and should not relinquish this accountability requirement. When the budgetary cycles for capitalized projects extend over a year or many years this calculation will be, and must be, done up front. It will not be precise simply because it is an estimate, but it must be done.

Note that I referenced *capitalized* projects. Many system maintenance organizations operate on an expensed basis. These organizations typically use a workflow throttling approach—they collect all of the system changes that are required and requested, they prioritize them according to urgency or value and they accept and tackle only as much work as they can do. When priorities and urgency change, what they choose to work on changes too. They are agile and they have been that way forever. But they are allowed to be because their budgetary cycle is continuous and expensed. Much of the conflict between resource management expectations and agile developer desires comes from this difference in the budgetary resourcing cycle. Another conflict comes simply from misunderstanding the nature of estimates and commitments.

Estimates are always uncertain; uncertainty is baked into the very definition of the word.

Estimation vs. Commitment

These are not the same, though in many organizations the difference is not recognized. An estimate is an uncertainty range, but we cannot commit to an uncertainty. We must tell the customer some date and we must budget some amount. So we need to turn an uncertain estimate range into a certain value. This value is the commitment.[1]

The commitment is the estimate PLUS a calculated reserve needed to pay for the risk inherent in the situation. Projects that have poorly defined requirements, whether due to market forces, customer uncertainty, or even low project skills, take on more risk and require more resources. A key diagnostic that risk is not being considered is when I hear people being instructed to "take out the fat" from an estimate. The word is loaded. It implies surplus unnecessary resources have been added (as indeed they might have been if people expect to be unjustly punished for the inherent risk). If we reframed the command to "take out the resources needed to deal with risk" we would be closer to reality. Projects that do this are fine—as long as the risk does not actually happen. When it does, and the project cannot recover because it does not have the necessary resources, it means we did not accurately calculate the commitment.

We do not need an accurate estimate as much as we need an accurate commitment.

Return and Cost Plus Risk

The commitment is (should be) the *risk-weighted* cost of the project. That is, the cost of the project plus the cost of risk. Both values are estimates and both values are uncertain, but they are essential to making good business decisions. On the other side of the equation, we should also calculate the risk-weighted value from this project and calculate the risk-weighted return, but that is topic for another time.

Any Road Will Do

Agile projects, like all projects, need to have some target end state toward which they can work and estimation is one of the business management elements of that. This does not mean that some work cannot go ahead—funding and building a prototype is an entirely appropriate way of both defining requirements and creating an estimate (though usually not a system). Without an up-front estimation process of some sort we would not have a clue about whether we should run a project at all, irrespective of whether we choose to run it as an agile project or something else. True, we may not be able to completely define the requirements up-front and in most cases we probably should not try. But without the guidance given by a well-managed estimation process and the explicit risk resourcing that produces a feasible commitment, companies would need to stock up on their supply of blank checks. We still need to estimate. Estimation is not evil. When done properly estimation is good. C

References

1. Armour, P.G. The inaccurate conception. *Commun. ACM 51*, 3 (Mar. 2008).
2. DeMarco, T. *Why Does Software Cost So Much?* Dorset House Publishing, NY, 1995.

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

DOI:10.1145/2542506

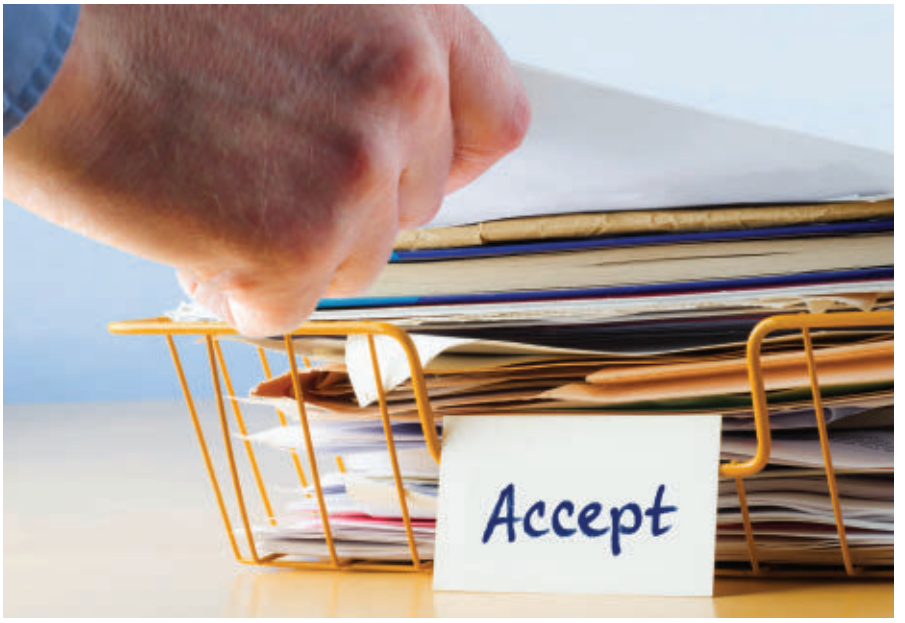
Doug Terry

Viewpoint
Publish Now, Judge Later

A proposal to address the problem of too many conference submissions and not enough time for reviewers to carefully evaluate each one.

Conferences in the computing field have large numbers of submissions, overworked and overly critical reviewers, and low acceptance rates. Conferences boast about their low acceptance rates as if this were the main metric for evaluating the conference's quality. With strict limits placed on the number of accepted papers, conference program committees face a daunting task in selecting the top papers, and even the best committees reject papers from which the community could benefit. Rejected papers get resubmitted many times over to different conferences before these papers are eventually accepted or the authors give up in frustration. Good ideas go unpublished or have their publication delayed, to the detriment of the research community. Poor papers receive little attention and do not get the constructive feedback necessary to improve the paper or the work.

Because reviewers approach their job knowing they must eventually reject four out of five submissions (or more), they often focus on finding reasons to reject a paper. Once they formulate such a reason, correctly or incorrectly, they pay less thought to the rest of the paper. They do not adequately consider whether the flaws could be corrected through modest revisions or whether the good points outweigh the bad. Papers with the potential for long-term impact get rejected in favor of papers with easily evaluated, hard to refute results. Program committees spend considerable time trying to agree on the best 20% of the papers that were submitted rather than providing comments to improve the papers for the good of all. Even if committees were able to perfectly order submissions according to quality, which they are not, papers that are close in quality may receive different outcomes since the line needs to be drawn somewhere. People do not always get the credit they deserve for inventing a new technique when their submission is rejected and some later work is published first.



A Proposal

My proposed solution is simple. Conferences should accept and publish all reasonable submissions. Some fields, such as physics, I am told, hold large annual conferences where anyone can talk about almost anything. I am not suggesting our conferences accept every submission. I believe computing conferences should enforce some standards for publication quality, but our current standards are far too stringent. We might argue about what constitutes a reasonable publication. Keeping in mind the main purpose of publication is to teach others, here is what I suggest.

A submission is "reasonable," and hence publishable, if it contains something new (a novel idea, new experimental result, validation of previous results, new way of explaining something, and so on), is based on sound methodology, explains the novelty in a clear enough manner for others to learn from it, and puts the new results in a proper context, that is, compares the results fairly to previous work. Rather than looking for reasons to reject a paper or spending time comparing papers, the role of conference reviewers is (a) to assess whether each submission is reasonable according to this criteria, and, perhaps more importantly, (b) to offer concrete suggestions for improvement. Any paper meeting this criteria should be accepted for publication, perhaps with shepherding to ensure that the reviewers' suggestions are properly followed.

Ultimately, papers will be judged in the fairness of time by accepted bibliometrics, such as citation counts, and, more importantly, by their impact on the field and on the industry. The importance of a published paper is often not known for many years. The "10 years after" or "hall of fame" awards should be used as the way to honor the best papers. These awards should be noted in the ACM Digital Library. Search engines, along with collaborative filtering and public recommendations, could direct researchers to high-quality, relevant work.

Practical Issues

What if a conference accepts more papers than can be presented during the length of the conference? In the steady state, this may not be a serious problem since there are lots of conferences and not that many new papers. If papers stop being submitted to (and rejected from) a half-dozen conferences, we will end up with far fewer submissions. To deal with large numbers of papers, conferences may need to have parallel sessions or shorter presentations or both. Personally, I am a fan of shorter presentations. An author should be able to present the key idea behind his or her work in 10–15 minutes and let people read the paper for more detail. Some papers could be presented as posters only, but I am not a fan of this approach. I would prefer to see all accepted papers treated equally. Let the community judge the papers.

How do authors decide where to submit their papers? Conferences will still have topics of focus. For example, we will still have conferences on databases, algorithms, systems, networks, and so forth. One additional criterion for acceptance is the paper fits the topical scope of the conference. Some papers may fit into multiple conferences. For example, a paper on distributed storage systems could be a database paper and a systems paper, that is, be suitable for presentation at SIGMOD or SOSP. In this case, since the criteria for accepting papers is the same for all conferences, it does not matter much to which conference the paper is submitted. In either case, assuming they are ACM conferences, the paper will end up in the Digital Library. Most likely, an author will submit his or her paper to the conference that attracts the community to which he mostly closely aligns, such as a conference that is sponsored by a Special Interest Group (SIG) to which he belongs. Low-quality conferences will likely go away, leaving one top conference in each technical area or for each technical community. To me, having fewer conferences would be a good thing.

What prevents people from submitting papers containing the "least publishable unit"? Authors can decide for themselves when they have a significant result they want to share with the community. Getting ideas and results published quickly is a good thing. There is no reason that someone should wait until they have a full paper's worth of results before submitting their work. The length of the paper can be commensurate with its contributions. People who submit lots of short papers with very marginal contributions risk harming their reputations and will likely receive fewer "test of time" awards than those that submit more major results. That may be sufficient incentive to discourage overly incremental submissions.

How would this affect journals? I suspect journal submissions would go up and more emphasis would be placed on journal publications. Journals would continue to have distinguished review boards that accept and

Calendar of Events

February 15–19
Computer Supported Cooperative Work,
Baltimore, MD,
Sponsored: SIGCHI,
Contact: Susan R. Fussell,
Email: sfussell@cornell.edu,
Phone: 607-255-1581

February 22–26
ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming,
Orlando, FL,
Sponsored: SIGPLAN,
Contact: Jose E. Moreira,
Email: jmoreira@us.ibm.com,
Phone: 914-525-6267

February 23–25
The 2014 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays,
Monterey, CA,
Sponsored: SIGDA,
Contact: Vaugn Timothy Betz,
Email: vaughnbetz@gmail.com,
Phone: 416-766-2197

February 24–27
19th International Conference on Intelligent User Interfaces,
Haifa, Israel,
Sponsored: SIGART, SIGCHI,
Contact: Tsvi Kuflik,
Email: tsvikak@is.haifa.ac.il

February 24–28
Seventh ACM International Conference on Web Search and Data Mining,
New York, NY,
Sponsored: SIGWEB, SIGIR, SIGKDD, SIGMOD,
Contact: Ben Carterette,
Email: Carteret@cis.udel.edu,
Phone: 302-31-3185

March 1–2
10th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments,
Salt Lake City, UT,
Sponsored: SIGPLAN, SIGOPS,
Contact: Martin Johannes Hirzel,
Email: hirzel@gmail.com

March 1–5
Architectural Support for Programming Languages and Operating Systems,
Salt Lake City, UT,
Sponsored: SIGPLAN, SIGOPS, and SIGARCH,
Contact: Rajeev Balasubrarmonian,
Email: Rajeev@cs.utah.edu

reject papers based on quality. Thus, a journal publication will be viewed as more prestigious than a conference paper. Papers with early results that are presented at conferences may later become journal articles with more substantial results, refined ideas, or practical experiences. Results from multiple conference papers may be combined into more comprehensive journal papers. This could make the publication practices for computing research more similar to those of other scientific disciplines.

Alternative Proposals

I am certainly not the first to observe flaws in our current publication practices or to suggest changes.[5,6] Attendees at a recent Dagstuhl Perspectives Workshop on the "Publication Culture in Computing Research" spent days debating alternatives. That workshop prompted this position statement. Others have suggested modifications to our publication processes, such as open access[1] and post-publication peer reviews,[3] and a number of these viewpoints have already appeared in *Communications*.[2,4,7] New services have been deployed for some communities, such as PubZone[a] which fosters public discussion of published papers in the database field. These practices and systems merit consideration, but are mostly orthogonal to what I propose.

Public websites, like the Computing Research Repository (CoRR),[b] have been established to encourage the rapid dissemination of new ideas. Authors may choose to make their papers immediately available by depositing them in such a repository. This approach addresses some of the problems that I raise, but differs in three fundamental ways. First, the authors do not get the thrill or experience of presenting their work in front of a live conference audience. Second, the deposited papers generally are later submitted for publication in a more established conference or journal. Therefore, concerns remain about repeated submissions and its load on reviewers. Third, and most importantly, the papers are not peer reviewed. My proposal retains pre-publication peer review. Thus, authors benefit from receiving constructive feedback that should be considered when revising their papers in advance of publication, and readers benefit from the knowledge that the work was vetted by a distinguished program committee.

I am certainly not the first to observe flaws in our current publication practices.

How to Get There

Adopting new publication policies is not simple. I do not expect established conferences to change their practices overnight. Conferences have a vested interest in protecting their hard-earned reputations by maintaining low acceptance rates. University computer science departments have succeeded at getting promotion committees to value conference publications, and are reluctant to make changes that might damage that position. Nevertheless, I believe that gradual steps are possible. As an encouraging trend, I know of a couple of recent systems conferences that accepted more papers than usual while continuing as single-track conferences. Serving as a program committee member for one of those conferences (MobiSys 2012), I observed firsthand the difficulty of getting reviewers to alter their mind-sets and accept even marginally more submissions.

One way to move forward is to establish new "high acceptance" conferences in addition to the existing "low acceptance" conferences. Adding more conferences is not a good long-term solution, but could nudge the community in the right direction, provide experimental data, and spark discussion. For example, last year SIGOPS held a new conference, the Conference on Timely Results in Operating Systems (TRIOS), in conjunction with its highly regarded Symposium on Operating Systems Principles (SOSP). This experimental conference accepted papers that were rejected from SOSP but still made a significant contribution. Lessons learned from this experiment are feeding into a broader discussion of publication practices in the SIGOPS community. TRIOS is providing insights into whether the community values conferences with less-constrained acceptance rates and whether authors will choose to present their work at such a conference or wait for publication opportunities that might look better on their résumés.

Conclusion

My main proposal is that conferences accept and publish any submission that contributes something new to our body of knowledge and that conveys its contribution in a clear and fair manner. The benefits of accepting any reasonable conference submission and abandoning low acceptance rates are clear:

- Research results get published in a timelier manner.
- Reviewers focus on providing constructive feedback.
- Program committees do not waste time reviewing the same submissions over and over again.
- Credit goes to those who first conceive of an idea and to groups that develop similar ideas in parallel.
- The community judges work by its long-term impact.

However, it does require a fundamental shift in how the research community, as well as tenure committees and other review boards, evaluates conference publications. I believe some kind of shift is needed. C

References

1. Beaudouin-Lafon, M. Open access to scientific publications. *Commun. ACM 53*, 2 (Feb. 2012).
2. Meyer, B., Choppy, C., Staunstrup, J., and van Leeuwen, J. Research evaluation for computer science. *Commun. ACM 52*, 4 (Apr. 2009).
3. Neylon, C. Reforming peer review. What are the practical steps?" (Mar. 8, 2011); http://cameronneylon.net/blog/reforming-peer-review-what-are-the-practical-steps/.
4. Roman, D. Scholarly publishing model needs an update. *Commun. ACM 54* (Jan. 2011).
5. Rosenberger, J. Should computer scientists change how they publish?, *BLOG@CACM* (July 29, 2012).
6. Vardi, M.Y. Revisiting the publication culture in computing research. *Commun. ACM 53*, 3 (Mar. 2010).
7. Wallach, D.S. Rebooting the CS publication process. *Commun. ACM 54*, 10 (Oct. 2011).

Doug Terry (terry@microsoft.com) is a Principal Researcher in the Microsoft Research Silicon Valley Lab, Mountain View, CA.

a PubZone Scientific Publication Discussion Forum; http://pubzone.org/.

b CoRR: Computing Research Repository; http://arxiv.org/corr/home.

LEARNING @ SCALE

MARCH 4–5 2014
ATLANTA, GEORGIA, USA
http://learningatscale.acm.org/

LEARNING @ SCALE

ACM will host the first **ACM Conference on Learning at Scale** to be held March 4-5, 2014 at the Hyatt Regency Atlanta, in Atlanta, Georgia, USA.

Inspired by the emergence of Massive Open Online Courses (MOOCs) and the shift in thinking about education, ACM created this conference as a new venue to explore how learning and teaching can change and improve when done "at scale."

ABOUT THE CONFERENCE

ACM Learning at Scale 2014 is the first in a new conference series intended to promote scientific exchange of interdisciplinary research at the intersection of the learning sciences and computer science.

Learning at Scale refers to new approaches for students to learn and for teachers to teach, when engaging hundreds or even thousands of students; be it face-to-face or remotely, synchronous or asynchronous.

Topics to be explored at the conference will include Usability Studies, Tools for Automated Feedback and Grading, Learning Analytics, Analysis of Log Data, Studies of Application of Existing Learning Theory, Investigation of Student Behavior and Correlation with Learning Outcomes, New Learning and Teaching Techniques at Scale.

ACM Learning at Scale 2014 will be co-located with **SIGCSE 2014**, the annual Technical Symposium of the ACM Special Interest Group on Computer Science Education (*http://sigcse2014.sigcse.org/*).

Together, these conferences will make for a great week spotlighting education!

COMMITTEE

GENERAL CHAIR

Mehran Sahami (Stanford University)

PROGRAM CHAIRS

Armando Fox (UC Berkeley)
Michelene T.H. Chi (Arizona State University)
Marti Hearst (UC Berkeley)

http://learningatscale.acm.org/

L@S Corporate Support Provided By:

Google Microsoft Research ORACLE ACADEMY

DOI:10.1145/2541883.2541893

Article development led by acmqueue
queue.acm.org

Dante's tale, as experienced by a software architect.

BY ALEX E. BELL

The Software Inferno

PREFACE: THE SOFTWARE INFERNO is a tale that parallels *The Inferno*, Part One of *The Divine Comedy* written by Dante Alighieri in the early 1300s. That literary masterpiece describes the condemnation and punishment faced by a variety of sinners in their hell-spent afterlives as recompense for atrocities committed during their earthly existences. The Software Inferno is a similar account, describing a journey where "sinners against software" are encountered amidst their torment, within their assigned areas of eternal condemnation, and paying their penance. In an attempt to preserve some of the original *Inferno*'s spirit and archaic prose, I have reused a few fragments of originally translated text taken from SparkNotes[3] and am grateful to include this classic verbiage.

The topology of the Software Inferno is captured in the accompanying figure and describes the setting for the sorrowful tale you are about to read.

The Journey Begins

Midway through our software project, I found myself in a dark wood, the right road lost. I cannot well recount how I entered it, so full was I of discontent at that point where I abandoned the true path. Yet, the true path is what I had hoped this journey would restore to my troubled soul, and will now tell of the things I have seen.

From amidst the darkness of the wood, I looked on high and saw a hill whose shoulders were cloaked by the rays of the sun, whose crest I was drawn to in hopes of reacquainting myself

ILLUSTRATION BY COLIN JOHNSON

with the true path. As I climbed higher and higher among the perilous ridges and furiously flowing waterways, I came across three beasts blocking my way. The beasts were a nasty and snarling lot, well known for tormenting not only strangers such as myself, but also for being at odds with one another. A difficult task mine would be to satisfy the coincident demands of the Cost, Schedule, and Quality beasts as a condition of passage. Given my wayward state of mind, I was able to successfully quell the wrath of only two of these beasts at any given time, and had little choice but to descend from the hill and ponder an alternate course for reacquainting myself with the true path.

While I was falling back to the low place from where I had started, before my eyes appeared one who through long study seemed to be a noble woman. "Have pity on me, I am lost!" I cried. "What so thou art, or shade, or real woman?" Responded she, "I was born Augusta Ada King, to parents Lord Byron and Anne Isabella Byron, resided at England, and became Augusta Ada King, Countess of Lovelace." Replied I, "Art thou then that Ada Lovelace and that fount from which flowed the first computer program?" Attested she did, "Indeed it is I."

I recounted my woeful tale to the Countess Lovelace about having lost the true path, my thus far futile attempts to rediscover it, and of the horrible beasts I had previously encountered while ascending the mountain trail. Said she to me, "Thee it behooves to hold another course." To my great relief, Lady Ada, the Countess of Lovelace, offered to serve as my guide in the quest of finding the true path but warned that the journey ahead would involve passing through the Software Inferno, a horrible place of eternal punishment for those who have committed sins in the realm of software development.

The countess bade me to follow, behind her I kept.

Ante-Inferno

"Leave every hope, ye who enter!" bore the inscription on the gate at the edge of the Software Inferno. Here were sighs, laments, and deep wailings resounding through the starless air. Strange tongues, horrible cries, words of woe, accents of anger, voices high

and hoarse, so were the unwelcoming environs of the Software Inferno.

As we approached the Inferno's edge, the countess said the woeful sounds we heard were those of software developers condemned to the Ante-Inferno. These developers were the wretches who could not decide whether they should for all time labor with conscience and morality using the skills of their trade for good, or to use those skills to promote evil, thievery, and illicit self-benefit. These were the hackers, the rogues, who either disrespected the true path by lucid choice, or failed to unambiguously endorse it by way of indecision. Both Heaven and Inferno have denied them entry, so they shall forever wallow here, tormented by ankle-biting bugs and typing on keyboards with the products of their strokes going nowhere, of importance to no one.

The countess beckoned that I follow her to the edge of a nearby river separating us from the first circle of the Software Inferno. Coming toward us in a boat, an old man with white ancient hair cried, "Woe to you, wicked souls! I come to ferry you to the other bank, into eternal torment, into heat and misery, into the Software Inferno!"

Into the boat stepped the countess, behind her I kept.

Canto 1: Limbo

The countess and I disembarked from the ferryman's boat and found ourselves at the edge of the Software Inferno's outer circle, also known as Limbo. The countess confessed that she herself was a Limbo inhabitant and had only been granted a short respite to act as my escort. She explained, "Through no fault of our own, we who have been condemned to Limbo are guilty only of being born too early in history, too young to have been aware of or to have properly paid homage to the visionaries responsible for shaping the software-engineering environment as you know it today."

Among the luminaries I recognized joining the countess in Limbo were the likes of George Boole (formalization of Boolean algebra), Gottlob Frege (first-order predicate calculus), and Grace Murray Hopper (Cobol inventor). They, too, suffered from the same fate as the countess: being born too early in time to have shown proper reverence to software-engineering demigods such as Alan Turing, Grady Booch, James Rumbaugh, Ivar Jacobsen, Alan Kay, Bran Selic, James Gosling, and others contributing to the basis of modern software engineering.

Presumably in exchange for having lived virtuous lives, the lone merciful consideration bestowed upon the blameless souls relegated to Limbo was life in the absence of misery such as that suffered by the unfortunates condemned to the more inward circles of the Software Inferno.

On led the countess, behind her I kept.

Canto 2: Lust

As the countess and I approached the Inferno's second circle, opine we did for the relative comfort of the circle we had just departed, as the inundating and blinding light emanating from the circle ahead bothered our eyes. It originally appeared as if the glow ahead was borne of a single source, but our ever-growing nearness showed that it was actually an assemblage of many individual light beams, each specifically focused on a single of the circle's many inhabitants.

The countess explained that the sins committed by these woeful souls were ones of lust. It was not a fleshly lust that these tormented souls were guilty of, but a lust for power, fame, fortune, and riches, at the betrayal of reason, commitment, and responsibility. These miserable wretches salaciously stoked only their selfish desires, regardless of the impact to those closest to them, without regard for their personal integrity, and in absence of good conscience. Among the condemned here were those who ignored or breached their wedding vows, denied their children an involved parent, or failed to honor their own parents in old age. Their blind pursuits were made in the name of the next deal, the next promotion, the next bonus, or anything else that was a means of accruing celebrity, power, or standing.

It was here, amidst the doom of the Inferno's second circle, where these obsessed souls would be eternally admonished with fitting punishment. A blinding spotlight focused on their each and every move, affording them the single-minded attention they had sought during their earthly days. So intense their illumination, the cylinders of light tormenting these souls prevented them from sleep and overwhelmed the natural blush of all that coexisted in the circle with them. For all of perpetuity, these ill-fated wretches were condemned to seeking shelter from the beams that followed their every move.

From the brightness led the countess, behind her I kept.

Canto 3: Gluttony

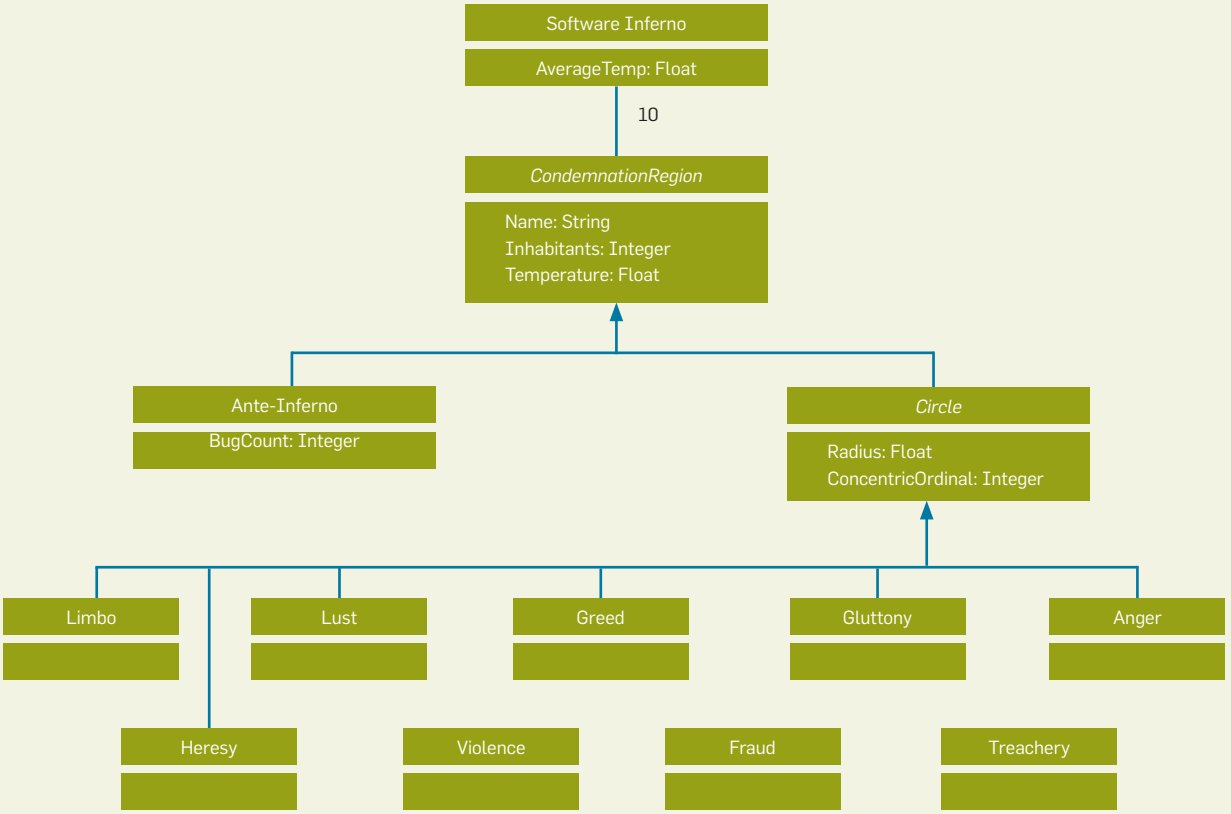
The countess and I slowly regained the comfort and use of our eyes as the extreme brightness of the previous circle waned. Coincidently, a new assault commenced on another of our senses in the form of horrible noises emanating from the Inferno's third circle, which lay just ahead. A curious sight awaited as we entered this circle: its inhabitants had the bodies of pigs, but they appeared to have retained their earthly heads from past days. The most repulsive sounds of slobbery and crudeness came from their mouths as they wallowed from trash heap to trash heap, eating all in sight, attempting to satisfy their unrelenting appetites with an absence of any and all grace.

The countess explained the souls doomed here were guilty of wasteful, excessive, and inappropriate consumption during their past lives. Their gluttony was largely stoked by immense wealth had suddenly fallen upon them as the result of IPOs, buyouts, and soaring stock prices. Fueled by their newly found prosperity, they gorged themselves with the finest foods, drank to excess, became dependent on prohibited substances, and contentedly watched their bellies grow fatter and fatter as badges of their success. The wealth that they had acquired robbed them of moderation, self-control, and any remembrance of those in the world who knew not from where there next meal might come.

So did the gluttonous spend eternity atoning for lifestyles of their earthly days: eating filth-ridden, rotten, and vermin-infested scraps of food that were unfit for beast, let alone human.

Away led the countess, behind her I kept.

UML representation of the Software Inferno.



Canto 4: Greed

The rhythmic, metallic, clinking sounds emanating from the Inferno's oncoming fourth circle were unlike anything my ears had heard before. The clamor became louder as the countess and I approached, and soon we were enlightened of its origin: pennies. Pennies were steadily falling from the sky, without end, often striking the heads of those wretches condemned to this fourth circle, and then clinking to the ground, striking other pennies that had fallen before them.

The Countess Ada explained the wretches condemned to the Inferno's fourth circle had demonstrated a habitual pattern of greed for wealth and material in their earthly days. Among the casualties of this greed included the programmers, designers, analysts, administrators, testers, and architects upon whose backs their wealth was gained, with nary a hint of appreciation offered for their hard work, loyalty, and dedication in return. The countess continued to describe that many of these greedy souls failed not only to justly reward their benefactors, but also to give back to community, to charity, or to educational institutions with even the smallest crumbs.

The woeful inhabitants of the fourth circle of the Software Inferno were without shelter, and were forever under the deluge of falling pennies interrupting any attempt to sleep or rest, providing them ample time to reflect on their greed of past times.

The countess bade me to follow, behind her I kept.

Canto 5: Anger

My still-ringing ears were gratified to have been spared the raucous din of falling pennies. But, the red glow and heat emanating from the nearing fifth circle did not bode well for the potential discomfort of other senses. As the countess and I approached, the heat became more difficult to bear, and the source revealed itself: innumerable piles of paper rubbish burning with flames shooting up high into the sky. The pitiful souls inhabiting this arid wasteland were desperate with thirst and appeared intent in its reprieve by dropping their buckets into the many wells dotting the terrain, amidst the burning piles, and hopeful of lapping any cool water that might be contained therein.

The countess explained the parched wretches condemned to the fifth circle were the angry, those who poisoned their software organizations from the inner cores. Their incessant vitriol, opinionated bullying, unending negativity, and caustic attitudes destroyed all prospect of harmony and collaboration in the workplace, creating only strife and misery for those around them. Not only did these toxic souls decline to apply their self-proclaimed wisdom to act in the place of the leaders whom they condemned, but they also undermined the efforts of anyone who attempted to assume those leadership roles themselves.

As more closely I watched, the angry who had hoped for cool water with which to slake their thirsts instead

spat the water out that they retrieved with their buckets, for it was poisoned, just as the workplaces they had poisoned in the past.

On led the countess, behind her I kept.

Canto 6: Heresy

The radiant heat of the previous circle soon yielded to metallic sounds that I believed to be shackles, clanking amidst irregular movement. As we approached the Inferno's sixth circle, the origin of these sounds became more clear. Their origin was from the many groupings of exactly nine people, chained to one other, with each individual intent on trudging in a direction and pace of his or her own choosing.

The countess explained these chaotically traveling souls were strongly at variance with well-established beliefs and laws of software engineering developed by experts of the related subject matter. Their unabashed contempt for universally accepted truths spawned decision making that wrought great damage upon software projects in their charge. Some challenged Fred Brooks' sacred counsel[1] in futile attempts to rise above their failings by adding new people with woefully insufficient qualification to rescue already-late projects. Others flaunted their derision by disregarding software design patterns sanctified by the Gang of Four,[2] instead opting for inelegance of their own in attempts to solve problems whose solutions were already proven, well known, and time honored.

The wretches condemned to this circle would spend the rest of eternity reliving the torment they had inflicted on others as the result of their heretical actions. As members of these shackled groups, they would be forever reminded how even the simplest of wills is difficult to execute in their number, and that nine people are not able to arrive at their desired destinations nine times faster than a single individual.

On marched the countess, behind her I kept.

Canto 7: Violence

As we approached the Inferno's next circle, the sounds of clanking shackles gave way to agonizing cries of torment that would suddenly and forcefully begin, but then gradually fade into

The countess explained the parched wretches condemned to the fifth circle were the angry, those who poisoned their software organizations from the inner cores.

moans of hopelessness and despair. Again and again, and much to my horror, our ears heard this cycle of misery repeat. I was certain that my eyes would soon fall upon a torture chamber overrun with miserable souls being tormented and made to suffer by horrible monsters, but observed nothing of the sort. Instead, I saw many people on the ground that were either asleep, writhing in pain, foaming at the mouth, with discolor of skin, or showed other symptoms of horrible pestilence.

The countess explained that these moaning scourges were condemned to the Inferno's seventh circle for committing acts of software violence against innocent and unsuspecting people. In front of our very eyes were the phishers, malware creators, and cybercriminals whose acts of hostility caused damage on the wealth, privacy, and general welfare of fellow netizens. Some had even inflicted great strife upon the unsuspecting and undeserving simply for sport, as hobby, merely to dispose of their restlessness.

More carefully I watched, and now understood the cause of their despair. While these wretches slept, becoming defenseless and unsuspecting, small flying worms appeared from the sky, alit on their skin, attached, and deposited toxins in them to inflict unspeakable pain and suffering when they awakened. Some attempted to avoid sleep with hope of avoiding the worms that wrought the virus and disease, but it was for naught. Comfort and refuge for these wretches was elusive for all of eternity as retribution for their sins.

Onward strode the countess, behind her I kept.

Canto 8: Fraud

As we approached the Inferno's eighth circle, the countess and I were besieged with an overwhelming stench that became more foul with each step taken. Which sins could have been committed by the miserable wretches here to deserve such reckoning? Explained the countess, "The souls condemned here are the frauds, liars, and deceitful: ones who feigned expertise, knowledge, and relevance of technology to others in their past lives, purely for their own advantage."

Ahead of us dwelt the self-proclaimed advocates, crusaders, and

evangelists of technologies such as UML, XML, Agile, MDA, object orientation, and the like. The fraudulent and deceptive were condemned here for having misdirected the adoption or usage strategies of technology on software projects in which they had had a role. These wretches willfully, and without apology, preyed upon the naïve and desperate, falsely convincing them that their pet technologies would bring relief to the torment-stricken for motivations of self-interest.

In the distance I spied a massive body of liquefied pulp, clearly being the source of the stench that fouled our nasal cavities. There were thousands of miserable souls wallowing in this morass, each struggling to stay above the surface of the simmering, bubbling, and putrid stew. Feeding this morass was a stream of material falling from the sky, in what seemed to be an endless supply. The countess explained that materials feeding the pulpy sewage in which these wretches floundered included unneeded UML diagrams, unachievable schedules, irrelevant training material, flowery PowerPoint charts, and other products the admonished had caused the creation of in their past lives. It was a fitting manner for the fraudulent and deceptive to spend eternity bathed in the stinking waste of their own creation.

On led the countess, behind her I kept.

Canto 9: Treachery

The Software Inferno's innermost circle lay ahead, situated atop a great mountain of substance and color whose like I before had not seen. Higher and higher the countess and I climbed, amidst a stench that my nostrils knew not could be more foul than the putrid stew we had left behind in the eighth circle below. When we could climb no more, before us labored a countless number of wretches, dumping shovelfuls of the mountain's body onto the souls of the eighth circle far below, providing the source of the noxious stew in which they wallowed. Said the countess to I, "The souls condemned to the ninth circle are guilty of sponsoring and acting upon errant guidance given by the fraudulent wretches stewing in the eighth circle with great impact to programs in their charge, without challenge, and in defiance of the more sage."

Great had been the struggle to bite my tongue thus far on the journey, but no longer was I able. With consent of the countess, I called to the nearest wretch having a load of foulness in his shovel, "Sir, which act of treachery did you commit to be condemned to this horrible place?" Replied he, "I am falsely accused of treachery, yet guilty only of being deceived by the fraudulent." Responded I, "Is it not treachery that you dismissed the counsel of your wisest sages, to the harm of your charge, because it came at the risk of forsaking the fruit and spoils promised by the fraudulent, upon which you relied for your own glory?"

Cried I to another miserable soul within earshot, "You sir, which treachery have you committed to be eternally condemned to a forsaken place such as this?" Replied he, "I am also falsely accused; my guilt is only failure to act against the angry." Responded I, "Is it not treachery for you to force your people to drink from a poisoned well and suffer its effects while you contentedly drink from your private one having fresh water?"

As fate would have it, not only were the treacherous forced to languish amidst the stench of products whose creation they had sanctioned, but their backs also eternally strained under their burden.

From the Software Inferno led the countess, behind her I kept.

The Truth Path

Thus completes the catalogue of evil filling the Software Inferno, of the people condemned therein, and of the admonishments cast upon them. I observed much suffering, horror, and despair in my travels but am now much the wiser for having done so and am confident this journey provided the enlightenment necessary to restore my trajectory back onto the true path.

My journey served as a reminder that simultaneously placating the beasts of three—Cost, Schedule, and Quality—is even more challenging when working amongst those who commit sins such as those condemned to the Software Inferno. In fact, I now believe my inability to secure safe passage beyond these temperamental beasts early in my journey was a reprimand for not having worked more diligently to help rid past projects of such sins and sinners.

Proactive leadership, value-based action, and balanced judgment are required to tame the beasts of three. The angry cannot be allowed to poison the workplace, the fraudulent cannot be allowed to derail the common good, and the treacherous cannot be allowed to lead with a wayward compass. Developing quality software, on schedule, and within budget is difficult enough without also having to deal with avoidable and crippling disturbances.

With my journey complete, I must now make haste and return to my earthly existence. Shout must I from the highest mountain all that the countess has shown me to be the true path. Leaders must heed the sage, create harmony, and actively rid workplaces of scourge. The self-absorbed must take inventory of their lives and readjust their priorities. The successful must remember those who help them succeed, and share fruit.

Many more will be my shouts as the result of what the countess has shown me to be good. But lament I do not having any special power to make well the ear of the deaf, to sway the stubborn, to soften the angry, or to enlighten the fool, because it is for souls such as these that the Software Inferno patiently awaits. C

Related articles on queue.acm.org

Death by UML Fever
Alex E. Bell
http://queue.acm.org/detail.cfm?id=984495

Coding for the Code
Friedrich Steimann and Thomas Kühne
http://queue.acm.org/detail.cfm?id=1113336

Software Development with Code Maps
Robert DeLine, Gina Venolia, and Kael Rowan
http://queue.acm.org/detail.cfm?id=1831329

References

1. Brooks, Jr., F.P. *The Mythical Man-Month.* Addison-Wesley, Reading, MA, 1975.
2. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. *Design Patterns.* Addison Wesley, Reading, MA, 1995.
3. SparkNotes Editors. SparkNote on *Inferno.* SparkNotes LLC, 2002; http://www.sparknotes.com/poetry/inferno/.

Alex Bell is a software architect with The Boeing Company. He is the author of "Death By Agile Fever," http://www.infoq.com/articles/death-by-agile-fever/.

DOI:10.1145/2541883.2541894

Article development led by acmqueue
queue.acm.org

Enterprise computing in the public cloud.

BY JASON LANGO

Toward Software-Defined SLAs

THE PUBLIC CLOUD has introduced new technology and architectures that could reshape enterprise computing. In particular, the public cloud is a new design center for enterprise applications, platform software, and services. API-driven orchestration of large-scale, on-demand resources is an important new design attribute, which differentiates public-cloud from conventional enterprise data-center infrastructure. Enterprise applications must adapt to the new public-cloud design center, but at the same time new software and system design patterns can add enterprise attributes and service levels to public cloud services.

This article contrasts modern enterprise computing against the new public-cloud design center and introduces the concept of software-defined service-level agreements (SD-SLAs) for the public cloud. How does the public cloud stack up against enterprise data centers and purpose-built systems? What are the unique challenges and opportunities for enterprise computing in the public cloud? How might the on-demand resources of large-scale public clouds be used to implement SD-SLAs? Some of these opportunities might also be beneficial for other public-cloud users such as consumer Web applications.

Today the dominant architectural model for enterprise computing is the purpose-built system in private data centers, engineered to deliver guaranteed service levels to enterprise applications. The architectural model presented by large-scale multitenant public clouds is quite different: applications and services are built as distributed systems on top of virtualized commodity resources. Many large-scale consumer Web companies have successfully delivered resilient and efficient applications using this model.

Getting enterprise applications into the public cloud is no easy task, but many companies are nonetheless interested in using cloud infrastructure broadly across their businesses, whether via public-cloud or private-cloud deployments. New levels of flexibility and automation promise to streamline IT operations. To become the primary computing platform for most applications, the public cloud needs to be a high-performance enterprise-class platform that can support business applications such as financial analysis, enterprise resource planning (ERP) systems, and supply chain management. Next, I look at practical systems considerations necessary for implementing enterprise cloud services.

Enterprise SLAs vs. Public-Cloud Design Center

Enterprise can be interpreted broadly as a business context requiring premium attributes such as high availability, security, reliability, and/or performance. This definition holds regardless of whether an application is legacy or new. For example, an enterprise analytical database might be implemented using a new scale-out architecture, and yet have enterprise requirements. Data security may be at a premium for either regulatory or busi-

PHOTOGRAPH FROM SHUTTERSTOCK.COM

ness reasons. Data integrity is at a premium because a mistaken business decision or financial result can cost the company real revenue or possibly even a loss in market value. Enterprise service levels are simultaneously of high business value and technically challenging to implement.

SLAs specify enterprise service-level requirements, often in the form of a legal contract between provider and consumer, with penalties for noncompliance. Concrete and measurable service-level objectives (SLOs) are individual metrics used to test that an SLA is being met. This distinction is important in the context of this article, which later identifies programmatically enforceable SLOs governed by a SD-SLA.

In this article, *public cloud* refers to a platform that deploys applications and services, with on-demand resources in a pool large enough to satisfy any foreseeable demand, run by a third-party cloud service provider (CSP). Many popular Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) providers meet this definition, including Amazon Web Services, Microsoft Azure, and Google Compute Engine. Cloud computing conventionally includes on-demand self-service, broad network access, resource pooling (aka multitenancy), rapid elasticity, and measured service.[12]

Unfortunately, there is a recognized gap between service levels the enterprise expects and what today's public cloud delivers. Current public-cloud SLAs are weak—generally providing 99.95% data-center availability and no guarantee on performance—and penalties are small.[4] Which ones matter? Why are they challenging? How can they be implemented? Let's take a long view on the advancement of both the public-cloud and enterprise infrastructure. While the public cloud is still undergoing rapid development and growth, it is possible to observe some trends.

Reliability and availability. The availability component of an enterprise SLA can be technically challenging. For example, a business-critical application might not tolerate more than five minutes of downtime per year, conforming to an availability SLO of 99.999% ("5 nines") uptime. In contrast, re-

sources in the public cloud have unit economics falling somewhere between enterprise and commodity hardware components, including relatively high expected failure rates. Amazon's virtual block devices, for example, have an advertised annual failure rate of 0.1%–0.5%, meaning up to 1 in 200 will fail annually.

Business-critical applications often have a low tolerance for application-level data inconsistency and zero tolerance for data corruption. Many enterprise applications may be reimplemented using an "eventual consistency" architecture to optimize both performance and availability at the cost of compensating for temporary inconsistency.[3] When the business risk or penalty is high enough, however, some enterprise applications prefer taking some downtime and/or data loss rather than delivering an incorrect result. If the availability SLO is stringent enough, it places pressure on software to implement rapid recovery to maintain the requisite amount of uptime.

Leading CSPs have pushed for developers to adopt new fault-tolerant software and system-design patterns, which make few assumptions about the reliability and availability of underlying infrastructure. The public-cloud design center encourages "designing for failure"[10] as part of normal operation to achieve high availability. This creates a need for fault-tolerant software to compensate for known unreliable infrastructure, metaphorically similar to how a RAID (redundant array of independent disks) compensates for unreliable physical media. Reliability and availability have become software problems. On the plus side, it is an opportunity to build more robust software.

Performance. Enterprise-application performance needs vary. End-user-facing applications might be managed to a specific response-time SLO, similar to a consumer Web application measured in fractions of a second. Important business applications such as ERP and financial analysis might be managed to both response time and throughput-oriented SLOs, supportive of specific business objectives such as overnight trading policy optimization.

In the public cloud, many performance challenges are byproducts of multitenancy. Physical resources behave as queuing systems: oversubscription of multitenant cloud infrastructure can cause large variability in available performance.[16] "Noisy neighbors" may be present regardless of whether storage is rotational or solid state, or whether networking is 1 gigabit or 100 gigabits. Computing oversubscription can also negatively impact I/O latency.[18] An operational trade-off exists between performance and cost. Multitenant public clouds allow for high utilization rates of physical infrastructure to optimize costs to the CSP, which may be passed on as lower prices. Unfortunately, performance of shared physical resources cannot be guaranteed at the lowest possible fixed cost. Performance of oversubscribed physical resources can fluctuate randomly but is "cheap," whereas performance of statically partitioned physical resources can be guaranteed but at a higher cost. Amazon Provisioned IOPS (I/O operations per second) is an example of this trade-off, where guaranteed performance comes with a roughly double increase in cost.[2]

Flexible use of virtual resources is a requirement in the public cloud, especially if performance is to be guaranteed. Distributed systems must be actively managed to achieve performance objectives. The advantage of on-demand resources is that they can be reconfigured on the fly, but this is also a major software challenge.

Security requirements vary by application category but generalize as risk management: the higher the business or regulatory value of an application or dataset, the more stringent the security requirements. In addition to avoiding denial of service, which aligns with availability, and avoiding "data leakage," there is also a desire to increase "mean time to compromise" by putting multiple layered security controls in place, in recognition that no individual system can be perfectly secure.

The public cloud is an interesting environment from an enterprise security perspective. On the one hand, a multitenant public cloud is considered a new and worrisome environment. On the other hand, the ability to impose logical security controls and automate policy management across running workloads presents an opportunity. Logical controls are more flexible, auditable, and enforceable than physical controls. Network access-control rules are a classic example of logical controls, which may now be applied directly to virtual machines rather than indirectly via physical switch ports. Logical segmentation is able to be provisioned dynamically and can shrink to fit the exact resources in a running workload and move when the workload moves.

The public cloud demands new security tools and techniques, requiring a rethinking of classic security techniques. There is a need for programmatically expressing security SLOs. User, application, and dataset-centric policy enforcement are worthy areas of further exploration toward the challenge of implementing higher-level security SLAs (for example, "users outside the finance group may not access financial data" and "data at rest must be reencrypted every two hours").

From Purpose-Built Systems to Distributed Systems

Enterprise data centers are typically optimized for a predetermined set of use cases. Purpose-built systems, such as the one described in Figure 1, are engineered to achieve specific service levels with a fixed price/performance via preintegrated components. These come in various form factors: hardware appliances, preintegrated racked systems, and more recently, virtual appliances and cloud appliances (providing an out-of-the-box private cloud with preconfigured SLAs). Vendors vertically integrate hardware and software components to provide service-level attributes (for example, guaranteed rate I/O, reconfiguration of physical resources, fault isolation, and so on). Higher-level SLAs are met by combining a vendor's deployment recommendations with best practices from performance and reliability engineering.

Purpose-built systems currently offer very high performance levels for workloads that require high-bandwidth internode communication. I/O-intensive data analytics are an example: achieving low response-time SLAs means that

sustained internode traffic in double- or triple-digit gigabytes per second will exceed the conventional 10-gigabit Ethernet commonly found in large-scale public-cloud environments.

Enterprise buyers might justify additional expense for specific use cases—for example, technical computing users paid for early access to GPGPUs (general-purpose computing on graphics processing units) for parallel computation, while data warehousing users paid for InfiniBand or proprietary Banyan networking for higher-bandwidth data movement. In practice, specialized technology such as GPGPU has been delivered in limited quantities and geographies in the public cloud, with expanded availability over time. It takes a premium to continually stay on the bleeding edge.

Static and integrated, meet dynamic and distributed. CSPs are continually improving their offerings. The hardware gap between purpose-built systems and the public cloud is closing. Public-cloud providers have created hardware designs tailored for large-scale deployment and operational efficiency, with the Open Compute Project as a popular example.[9] The economic incentive for CSPs is clear: more use cases means more revenue. Amazon Web Services has been increasing instance (VM) performance for some time, motivated lately by business analytics (Amazon Redshift). This trend will likely continue because of the practical benefits of right-sizing virtual machine resources—for example, simple scalability issues (Amdahl's Law), reduced cost of data movement between nodes, or price/performance/power efficiency. Additionally, some CSPs might acquire purpose-built systems that provide guaranteed service levels, such as cloud appliances.

The public cloud is dynamic and distributed, in contrast to a purpose-built system, which is static and integrated. The CSP's virtualized resources are optimized for automation, cost, and scale, but the CSP also owns the platform and hardware-abstraction layers. Abstraction is a challenge in providing higher-level SLAs—it is difficult to guarantee service levels, not knowing which virtual resources are collocated within the same performance and failure domains.

Enterprise infrastructure has an opportunity to transform. New enterprise applications are being written against cloud-friendly software platforms such as Cloud Foundry and Hadoop. Implicitly, in the challenging effort of implementing highly available distributed systems, applications will be made robust against component failures, and the need for highly available infrastructure will diminish over time. Moreover, CSPs and platform services can help accelerate this transition. Microsoft Azure, for example, makes failure domains visible to distributed applications: a workload can allocate nodes from independent failure domains within the same data center. SLAs can be delivered in software on the public cloud, providing enterprise attributes and service levels to enterprise applications.

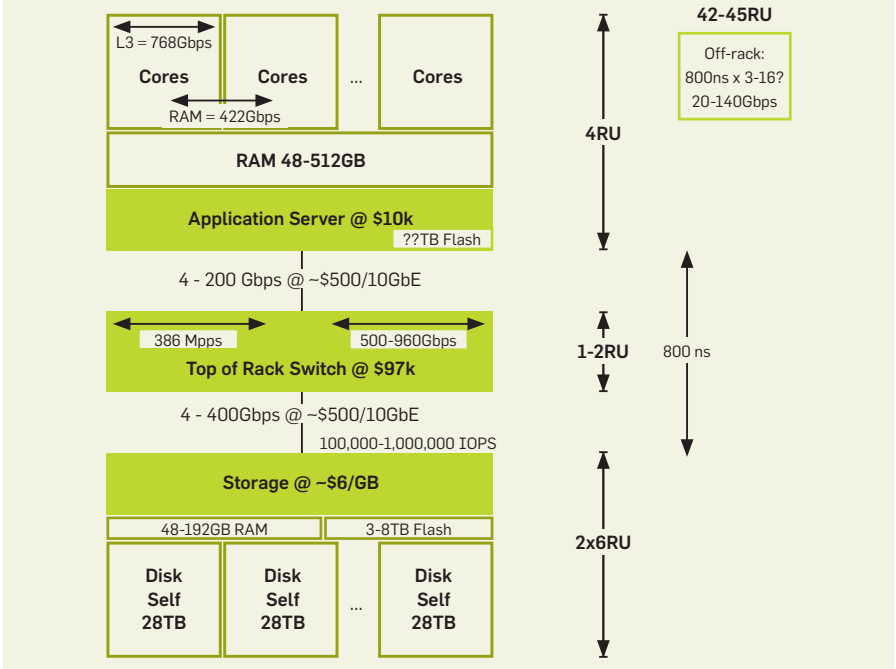
Software-Defined SLAs

On-demand resources in the public cloud are effectively infinite relative to the needs of the enterprise consumer. To appreciate this, it helps to get a sense of scale. Although rarely publicly disclosed, individual public CSPs have server counts conservatively estimated to be in the hundreds of thousands and growing rapidly.[14] That is already at least one order of magnitude larger than a reasonably large enterprise data center with tens of thousands of servers. At that scale, it is possible for an entire enterprise data center to fit within the CSP's idle on-demand capacity. In contrast to the millions of end users on a large-scale website, a population of 50,000 end users is a large number for an enterprise-business, custom in-house departmental, batch-processing, or analytical application.

This new design center fundamentally alters an architectural assumption in today's enterprise applications and infrastructure: the resource envelope is no longer fixed as in purpose-built systems or capacity-managed by central IT. Even additional CPUs and RAM can now be logically provisioned by enterprise applications and platform services at runtime, either directly or indirectly, by launching new virtual machines. The resource envelope is limited only by budget, but software has to be designed for the public cloud in order to exploit this.

While limited SLAs are available from the CSP, application and platform software components are generally required to provide guarantees

Figure 1. Enterprise rack diagram (ballpark list prices and specs are compiled from public data sources).



around application characteristics such as performance, resiliency, availability, and cost. Because of the challenges associated with multitenancy, public-cloud applications currently make few assumptions about the infrastructure underneath them. They are built to tolerate arbitrary failures by design and implement their own SLAs. There is an opportunity to create new architectural design patterns to help systematically solve some of these problems and allow for reusable components.

SD-SLAs are expected to increase in platform software components and cloud services optimized for the public-cloud design center. The next section provides examples, implementation considerations, and limitations and future opportunities.

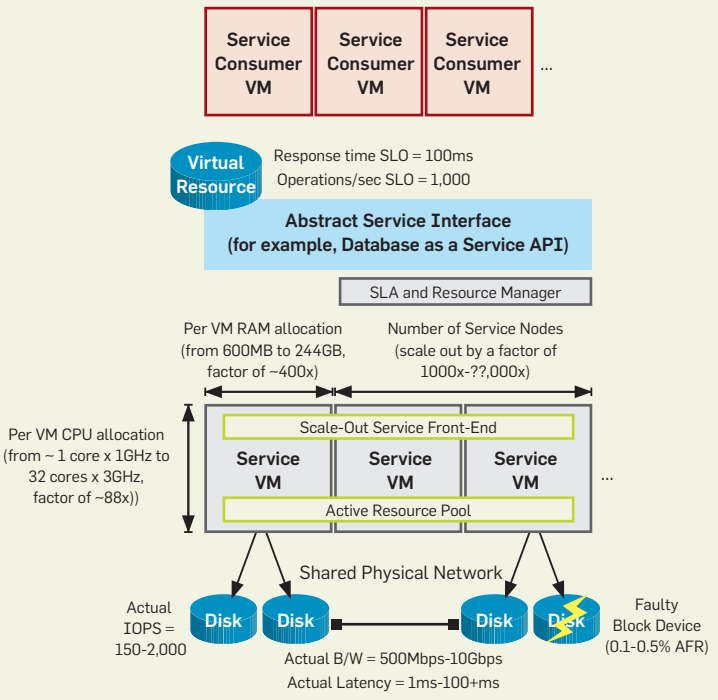
SD-SLAs offer a new design pattern that formalizes SLAs and SLOs as configurable parameters of public-cloud software components. Those components then manage underlying resources to meet specific measurable SLO requirements. With on-demand resources, a software systems layer can be implemented to meet some SLOs, which previously required planning, static partitioning, and overprovisioning of resources. Cloud service APIs may then begin to incorporate SD-SLAs as runtime configurations.

Programmatic SLOs within an SD-SLA might specify metrics for fundamental service levels such as response times, I/O-throughput, and availability. They might also specify abstract but measurable attributes such as geographic or workload placement constraints. Some examples: Amazon's service-oriented architecture featured a data service managed to a real-time SLA, which was dynamically sized and load-balanced to "provide a response within 300ms for 99.9% of its requests."[8] Amazon Provisioned IOPS allows for a given number of I/O operations per second to be configured per storage volume.

Many interesting targets for SD-SLAs are presented in the *ACM Queue* article, "There's Just No Getting Around It: You're Building a Distributed System,"[6] which also describes the challenge of building real-world distributed systems.

SD-SLAs should be vendor and technology independent, specified in logical units, and objectively measurable—for example, configure a desired number of I/O operations per second, as opposed to the number of devices necessary to achieve it; or an amount of bandwidth between nodes, as opposed to a physical topology.

Figure 2. Software-defined SLA in a public-cloud service.



Implementation considerations. SD-SLAs must necessarily be implemented in a distributed system in the public-cloud design center: for runtime-configurable SLOs to scale out; for high availability and fault tolerance; and to use on-demand compute and I/O resources.

First, consider this simple example: a reconfigurable I/O-throughput SLO guaranteeing some number of IOPS in the context of a distributed key-value store (see Figure 2). Assume the key-value store uses N-way replication with quorum-like consistency, such as in Dynamo, and that underlying storage volumes support a configurable performance capacity, such as in Amazon Provisioned IOPS. Given an initial configuration for I/O-throughput T, an SD-SLA-aware resource manager would allocate volumes sufficient to provide the desired aggregate I/O capacity. Conservatively and suboptimally, let's assume it allocates $T \times N$ IOPS to each volume, as each `get()` operation generates N concurrent I/O requests. In this example, the SD-SLA-aware resource manager could treat both SLO reconfiguration and poor-performing volume scenarios as a standard replica failure/replacement, providing automatic reconfiguration without further complicating the system with additional data copy code paths. In the event the I/O-throughput SLO is reconfigured to T', new volumes would be allocated at $T' \times N$ IOPS and old volumes failed out, until the system converges to T' aggregate I/O-throughput capacity. In the interim, a weighted I/O distribution policy might be used to maximize I/O throughput. In the real world, further performance and cost optimization would be required, and more sophisticated algorithms could be considered, such as an erasure coding instead of simple replication.

Given the challenge of distributed system development, a one-size-fits-all SD-SLA implementation is unlikely. A variety of programmatic SLOs may be implemented in application services, platform software components, or the CSP itself. The specific application context determines which

components are appropriate for a given use case. As both the public cloud and enterprise applications are moving targets, the industry is likely to continue iterating on which attributes are provided by the CSP, versus application, versus software components and services in between.

Runtime reconfiguration for SD-SLAs is challenging. QoS (quality of service) techniques such as I/O scheduling and admission control are necessary but not sufficient. Application- or service-specific implementation is necessary for dynamically provisioning RAM, CPU, and storage resources to meet changing SLOs or to meet SLOs in the presence of changing environmental conditions. The value of SD-SLAs, however, may justify significant engineering effort and cost. An example is the implementation of peer-to-peer object storage to allow for more fluid use of underlying resources, including the runtime replacement of compute nodes and flexible placement of data. Some SD-SLA implementations may use closed-loop adjustments from control theory.[11] Runtime reconfiguration may go hand in hand with resiliency to failure, as component replacement, initial configuration, and runtime adjustment may all be managed in a similar application-specific manner.

Placement of computation and data must be considered for performance and data-availability SD-SLAs. Collocation of computation and data can alleviate some performance issues associated with multitenant networking. Examples include flexible movement of computation and data implemented in Hadoop, Dryad, and CIEL; placement-related SLOs implemented in Microsoft Azure and Amazon Web Services (Affinity Groups and Placement Groups, respectively); data availability SLOs, specifying geographic placement and minimum number of replicas, implemented in Google Spanner.[7]

Tagging may be used in general to identify resources subject to SD-SLAs and specifically to implement security SLOs. In addition to resource tagging supported natively by CSPs, host-based virtual networking and OpenFlow offer further opportunities to tag users and groups in active network flows, similar to Cisco TrustSec and IEEE 802.1AE (the MAC security standard, also known as MACsec). Security SLOs may be implemented by associating user and group tags with access controls. Similarly, dataset-level tagging in storage service metadata assists in the implementation of dataset-level SLOs (for example, data availability, replication, access control, and encryption key management policy).

On-demand optimization. Even with the sophisticated tools and techniques around purpose-built systems, overprovisioning is the de facto standard method for guaranteeing service levels across the lifetime of a system. The entire cost of a purpose-built system must be paid up front, including the overhead of overprovisioning to meet SLAs and accommodate increasing usage over time. In contrast, the on-demand resources in the public cloud can be allocated and freed as needed, and thus may be billed according to actual use. This is an opportunity for the public cloud to outperform purpose-built systems in terms of operational efficiency for variable workloads.

Costs and resource allocation required to meet an SD-SLA may be tuned to optimize operational efficiency. Given that variable resources may be required to achieve different SLOs, a given SD-SLA may come associated with a cost function. Here are two fundamental theorems for the economics of SD-SLAs: (1) a change in any SLO must always be traded against cost as a random variable; and (2) in the face of changing underlying conditions (for example, unpredictable multitenant resources), cost is a random variable even when all other SLOs are fixed.

Programmatic cost modeling[13] and optimization[15] are new themes in public-cloud research, and work is ongoing.

Limitations and future opportunities. Unsurprisingly, there are both theoretical and practical limitations to SD-SLAs. Since cost is always a system-level parameter that must be managed, some combinations may not work. An invalid combination, for example, would be if an application demands one million IOPS with 1ms worst-case response time for a cost that is lower than the cost of the physical systems necessary to deliver this real-time SLA. Even given unlimited cost, some SLOs may be physically impossible to achieve (for example, a bandwidth greater than physical capacity of the underlying CSP or resource allocation faster than the underlying CSP is capable of providing it). Moreover, a poorly designed cloud service may not be amenable to SD-SLAs—for example, if fundamental operations are serialized, then they cannot be programmatically scaled out and up to satisfy an SD-SLA.

With SD-SLAs, there are further opportunities to move to a continuous model for many important background processes, which previously needed to be scheduled because of the constraint of fixed resources. Consider that an enterprise-storage or database system, rather than trusting underlying physical storage controllers, might have a software process that scans physical media to ensure that latent bit errors are corrected promptly. Since this process is potentially disruptive to normal operation in a system with fixed compute and I/O resources, the typical approach is to run it outside of business hours, perhaps on weekends every two to four weeks. Future cloud services with SD-SLAs might be designed to allow important background processes to run continuously without impacting front-end service levels delivered to the application, since both the front-end service and continuous background processes may have independent programmatic SLOs that scale out using on-demand resources.

Dynamic resource management is an area where competition between CSPs may unlock new opportunities (for example, "allocate a VM with a specific amount of nonvolatile RAM" or "add two more CPUs to this running VM"). Modern hypervisors already support this. Physical attributes can be disaggregated into individually consumable units. For example, compute resources can be allocated independently of I/O, I/O-throughput independent of capacity, and CPU and RAM independently of each other. This weakens the vertical-integration advantage of purpose-built systems. Amazon has approached this issue by offering a wide inventory of

VM types,[1] although finding the right combination of CPU and RAM may still involve overprovisioning one or the other.

Enterprise macro-benchmarks must be tailored to the new public-cloud design center. Much effort has gone into rigorous infrastructure benchmarks such as SPC-117 in the storage arena; however, the public cloud has introduced a fundamental economic shift—price/performance metrics need to factor in workload runtime. Thanks to the on-demand nature of the public cloud, price is a function of allocated resources over time, measured in hours or days since a workload started running, as opposed to a standard three-year life cycle of enterprise hardware. With SD-SLAs, allocated resources vary with time, front-end load, and whatever else is necessary to meet application SLAs. On the flip side, an I/O benchmark implemented in a massive RAM cache will yield stunning numbers, but price/performance must still be captured for this benchmark to be relevant. Further industry effort is necessary to evolve enterprise macro-benchmarks for the public cloud and SD-SLAs.

It is also natural to ask whether SD-SLAs are being met consistently. There are further opportunities to implement programmatic SD-SLA validation via automated test infrastructure and analytics.[5] This offers the opportunity for third-party validation of SLAs and assessing penalties appropriately.

Further industry and academic efforts can lead to fully flushing out the limits of SD-SLAs. It would be worth seeing how far we can go along these lines, perhaps one day getting close enough to approximate: "What application response time are you looking for? Here is what it will cost you."

Public Cloud Transcendent

The public cloud presents an opportunity to reimagine enterprise computing. It will be a rewarding journey for public-cloud services to take on the bulk of enterprise-computing use cases. As in past transitions, the transformation of enterprise applications from one model to the next can proceed incrementally, starting with noncritical applications and building upward as the ecosystem matures. The wheels are already in motion.

It is remarkable that a seven-year-old technology can be judged optimistically against the entire progress of enterprise infrastructure in the past 20–30 years. The pace of public-cloud innovation is relentless. A lot of energy and capital continues to pour into public-cloud infrastructure. Today, the public cloud is a multibillion-dollar market and growing rapidly. Any or all of today's issues could be gone in the blink of an eye. Enterprise platforms have historically seen radical shifts in structure as a result of the changing economics of computing—from the mainframe to client-server era. We are in the midst of another industry transformation.

Future enterprise applications and infrastructure may be built as distributed systems with reusable platform software components focused on the public cloud. This can assist information technology professionals and application developers in deploying fast and reliable applications without having to reinvent the wheel each time. Some enterprise features associated with reliability, availability, security, and serviceability could run continuously in this model. Runtime configuration of SD-SLAs provides an opportunity to manage based on the exact performance indicators that people want, as opposed to physical characteristics such as raw hardware or prepackaged SLAs. Enterprise applications can harness the scale, efficiency, and rapidly evolving hardware and operational advances of large-scale CSPs. These are all significant opportunities, not available in purpose-built systems but enabled by the large-scale, on-demand resources of the public cloud.

All engineers and IT professionals would be wise to learn about the public cloud and capitalize on these trends and opportunities, whether at their current job or the next. The public cloud is defining the shape of new software—from applications to infrastructure. It is our future. C

Related articles on queue.acm.org

Why Cloud Computing Will Never Be Free
Dave Durkee
http://queue.acm.org/detail.cfm?id=1772130

Condos and Clouds
Pat Helland
http://queue.acm.org/detail.cfm?id=2398392

There's Just No Getting Around It: You're Building a Distributed System
Mark Cavage
http://queue.acm.org/detail.cfm?id=2482856

References

1. Amazon Web Services. Amazon EC2 instances, 2013; http://aws.amazon.com/ec2/instance-types/.
2. Amazon Web Services. Amazon Elastic Block Store (EBS), 2013; http://aws.amazon.com/ebs/.
3. Bailis, P. and Ghodsi, A. Eventual consistency today: limitations, extensions, and beyond. *ACM Queue 11*, 3 (2013); http://queue.acm.org/detail.cfm?id=2462076.
4. Baset, S.A. Cloud SLAs: Present and future. *ACM SIGOPS Operating Systems Review 46*, 2 (2012), 57–66.
5. Bouchenak, S., Chockler, G., Chockler, H., Gheorghe, G., Santos, N., and Shraer, A. Verifying cloud services: present and future. *ACM SIGOPS Operating Systems Review 47*, 2 (2013), 6–19.
6. Cavage, M. There's just no getting around it: you're building a distributed system. *ACM Queue 11*, 4 (2013); http://queue.acm.org/detail.cfm?id=2482856.
7. Corbett, J.C. et al. Spanner: Google's globally distributed database. In *Proceedings of the 10th Usenix Conference on Operating Systems Design and Implementation*, 2012, 251-264.
8. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P. and Vogels, W. Dynamo: Amazon's highly available key-value store. *Proceedings of the 21st ACM SIGOPS Symposium on Operating Systems Principles*, (2007), 205–220.
9. Facebook. *Open Compute Project*, 2011; http://www.opencompute.org/.
10. Hamilton, J. On designing and deploying Internet-scale services. In *Proceedings of the 21st Conference on Large Installation System Administration*, 2007.
11. Hellerstein, J.L. Engineering autonomic systems. In *Proceedings of the 6th International Conference on Autonomic Computing*, 2009, 75–76.
12. Mell, P. and Grance, T. The NIST definition of cloud computing. National Institute of Standards and Technology Special Publication, 2011 800–145; http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.
13. Mian, R., Martin, P., Zulkernine, F. and Vazquez-Poletti, J.L. Estimating resource costs of data-intensive workloads in public clouds. In *Proceedings of the 10th International Workshop on Middleware for Grids*, Clouds and e-Science, 2012.
14. Netcraft. Amazon Web Services' growth unrelenting (May 2013); http://news.netcraft.com/archives/2013/05/20/amazon-web-services-growth-unrelenting.html.
15. Ou, Z., Zhuang, H., Nurminen, J. K., Ylä-Jääski, A., and Hui, P. Exploiting hardware heterogeneity within the same instance type of Amazon EC2. In *Proceedings of the 4th Usenix Workshop on Hot Topics in Cloud Computing* (HotCloud), 2012.
16. Schad, J., Dittrich, J. and Quiané-Ruiz, J.-A. Runtime measurements in the cloud: observing, analyzing, and reducing variance. In *Proceedings of the Very Large Data Base Endowment 3*, 1-2 (2010), 460–471.
17. Storage Performance Council. *SPC Specifications*, 2013; http://www.storageperformance.org/specs.
18. Xu, Y., Musgrave, Z., Noble, B. and Bailey, M. Bobtail: Avoiding long tails in the cloud. *Proceedings of the 10th Usenix Conference on Networked Systems Design and Implementation*, 2013, 329–342.

Jason Lango is co-founder and CTO of Bracket Computing, an enterprise cloud computing company he started while he was Entrepreneur in Residence at Sutter Hill Ventures. Previously, he was Principal Engineer at Cisco and a senior engineer at NetApp. Follow his blog at http://lastbusinessmachine.com.

DOI:10.1145/2541883.2541895

Article development led by acmqueue
queue.acm.org

What if all the software layers in a virtual appliance were compiled within the same safe, high-level language framework?

BY ANIL MADHAVAPEDDY AND DAVID J. SCOTT

Unikernels: The Rise of the Virtual Library Operating System

CLOUD COMPUTING HAS been pioneering the business of renting computing resources in large data centers to multiple (and possibly competing) tenants. The basic enabling technology for the cloud is operating-system virtualization such as Xen[1] or VMWare, which allows customers to multiplex virtual machines (VMs) on a shared cluster of physical machines. Each VM presents as a self-contained computer, booting a standard operating-system kernel and running unmodified applications just as if it were executing on a physical machine.

A key driver to the growth of cloud computing in the early days was *server consolidation*. Existing applications were often installed on physical hosts that were individually underutilized, and virtualization made it feasible to pack them onto fewer hosts without requiring any modifications or code recompilation. VMs are also managed via software APIs rather than physical actions. They can be centrally backed up and migrated across different physical hosts without interrupting service. Today commercial providers such as Amazon and Rackspace maintain vast data centers that host millions of VMs. These cloud providers relieve their customers of the burden of managing data centers and achieve economies of scale, thereby lowering costs.

While operating-system virtualization is undeniably useful, it adds yet another layer to an already highly layered software stack now including: support for old physical protocols (for example, disk standards developed

in the 1980s such as IDE); irrelevant optimizations (for example, disk elevator algorithms on SSD drives); backward-compatible interfaces (for example, Posix); user-space processes and threads (in addition to VMs on a hypervisor); and managed-code runtimes (for example, OCaml, .NET, or Java). All of these layers sit beneath the *application code*. Are we really doomed to adding new layers of indirection and abstraction every few years, leaving future generations of programmers to become virtual archaeologists as they dig through hundreds of layers of software emulation to debug even the simplest applications?[5,18]

This problem has received a lot of thought at the University of Cambridge, both at the Computer Laboratory (where the Xen hypervisor originated in 2003) and within the Xen Project (custodian of the hypervisor that now powers the public cloud via companies such as Amazon and Rackspace). The solution—dubbed *MirageOS*—has its ideas rooted in research concepts that have been around for decades but are only now viable to deploy at scale since the availability of cloud-computing resources has become more widespread.

The goal of MirageOS is to restructure entire VMs—including all kernel and user-space code—into more modular components that are flexible, secure, and reusable in the style of a library operating system. What would the benefits be if *all* the software layers in an appliance could be compiled within the same high-level language framework instead of dynamically assembling them on every boot? First, some background information about appliances, library operating systems, and type-safe programming languages.

Figure 1. Software layers and a stand-alone kernel compilation.

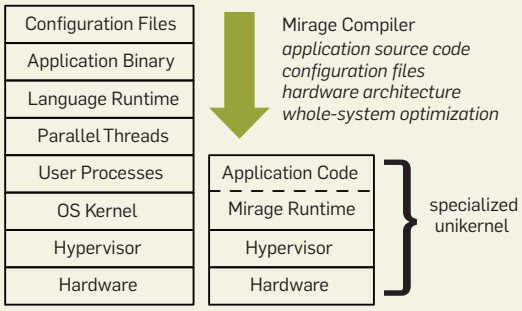
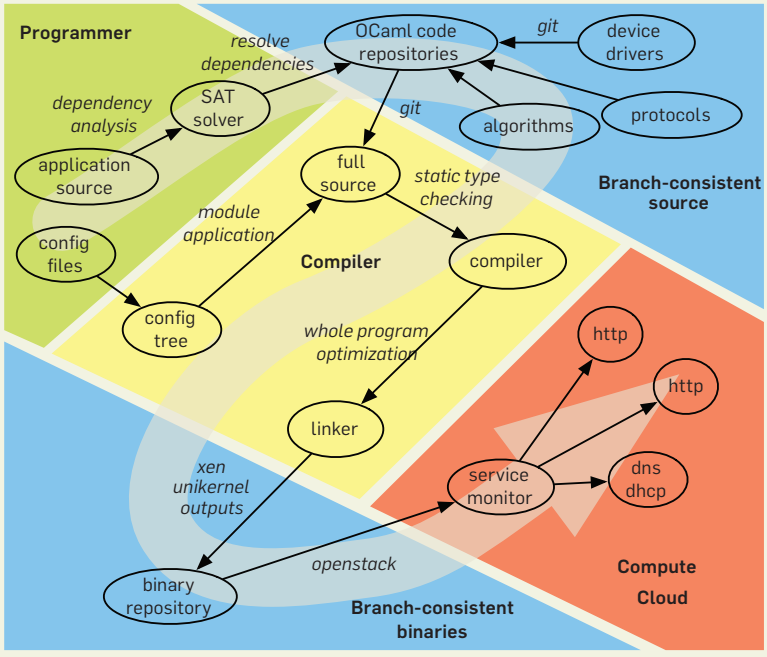


Figure 2. Logical workflow in MirageOS.



The shift to single-purpose appliances. A typical VM running on the cloud today contains a full operating-system image: a kernel such as Linux or Windows hosting a primary application running in user space (for example, MySQL or Apache), along with secondary services (for example, syslog or NTP) running concurrently. The generic software within each VM is initialized every time the VM is booted by reading configuration files from storage.

Despite containing many flexible layers of software, most deployed VMs ultimately perform a single function such as acting as a database or Web server. The shift toward single-purpose VMs is a reflection of just how easy it has become to deploy a new virtual computer on demand. Even a decade ago, it would have taken more time and money to deploy a single (physical) machine instance, so the single machine would need to run multiple end-user applications and therefore be carefully configured to isolate the constituent services and users from each other.

The software layers that form a VM have not yet caught up to this trend, and this represents a real opportunity for optimization—not only in terms of performance by adapting the appliance to its task, but also for improving security by eliminating redundant functionality and reducing the attack surface of services running on the public cloud. Doing so statically is a challenge, however, because of the structure of existing operating systems.

Limitations of current operating systems. The modern hypervisor provides a resource abstraction that can be scaled dynamically—both vertically by adding memory and cores, and horizontally by spawning more VMs. Many applications and operating systems cannot fully utilize this capability since they were designed before modern hypervisors came about (and

the physical analogues such as memory hotplug were never ubiquitous in commodity hardware). Often, external application-level load balancers are added to traditional applications running in VMs in order to make the service respond *elastically* by spawning new VMs when load increases. Traditional systems, however, are not optimized for size or boot time (Windows might apply a number of patches at boot time, for example), so the load balancer must compensate by keeping idle VMs around to deal with load spikes, wasting resources and money.

Why couldn't these problems with operating systems simply be fixed? Modern operating systems are intended to remain resolutely general purpose to solve problems for a wide audience. For example, Linux runs on an incredibly diverse set of platforms, from low-power mobile devices to high-end servers powering vast data centers. Compromising this flexibility simply to help one class of users improve application performance would not be acceptable.

On the other hand, a specialized server appliance no longer requires an OS to act as a resource multiplexer since the hypervisor can do this at a lower level. One obvious problem with this approach is that most existing code presumes the existence of large but rather calcified interfaces such as POSIX or the Win32 API. Another potential problem is that conventional operating systems provide services such as a TCP/IP stack for communication and a file-system interface for storing persistent data: in our brave new world, where would these come from?

The MirageOS architecture—dubbed *unikernels*—is outlined in Figure 1. Unikernels are specialized OS kernels that are written in a high-level language and act as individual software components. A full application (or *appliance*) consists of a set of running unikernels working together as a distributed system. MirageOS is based on the OCaml (http://ocaml.org) language and emits unikernels that run on the Xen hypervisor. To explain how it works, let's look at a radical operating-system architecture from the 1990s that was clearly ahead of its time.

Library operating system. This is not the first time people have asked these existential questions about operating systems. Several research groups have proposed operating-system designs based on an architecture known as a *library operating system* (or libOS). The first such systems were Exokernel[6] and Nemesis[10] in the late 1990s. In a libOS, protection boundaries are pushed to the lowest hardware layers, resulting in: a set of *libraries* that implement *mechanisms*, such as those needed to drive hardware or talk network protocols; and a set of *policies* that enforce access control and isolation in the application layer.

The goal of MirageOS is to restructure entire VMs—including all kernel and user-space code—into more modular components that are flexible, secure, and reusable in the style of a library operating system.

The libOS architecture has several advantages over more conventional designs. For applications where performance—and especially *predictable* performance—is required, a libOS wins by allowing applications to access hardware resources directly without having to make repeated privilege transitions to move data between user space and kernel space. The libOS does not have a central networking service into which both high-priority network packets (such as those from a videoconference call) and low-priority packets (such as from a background file download) are forced to mix and interfere. Instead, libOS applications have entirely separate queues, and packets mix together only when they arrive at the network device.

The libOS architecture has two big drawbacks. First, running multiple applications side by side with strong resource isolation is tricky (although Nemesis did an admirable job of minimizing crosstalk between interactive applications). Second, device drivers must be rewritten to fit the new model. The fast-moving world of commodity PC hardware meant that, no matter how many graduate students were tasked to write drivers, any research libOS prototype was doomed to become obsolete in a few short years. This approach worked only in the real-time operating-system space (for example, VxWorks) where hardware support is narrower.

Happily, OS virtualization overcomes these drawbacks on commodity hardware. A modern hypervisor provides VMs with CPU time and strongly isolated virtual devices for networking, block storage, USB, and PCI bridges. A libOS running as a VM

needs to implement only drivers for these virtual hardware devices and can depend on the hypervisor to drive the real physical hardware. Isolation between libOS applications can be achieved at low cost simply by using the hypervisor to spawn a fresh VM for each distinct application, leaving each VM free to be extremely specialized to its particular purpose. The hypervisor layer imposes a much simpler, less fine-grained policy than a conventional operating system, since it just provides a low-level interface consisting of virtual CPUs and memory pages, rather than the process and file-oriented architecture found in conventional operating systems.

Although OS virtualization has made the libOS possible without needing an army of device-driver writers, *protocol libraries* are still needed to replace the *services* of a traditional operating system. Modern kernels are written in C, which excels at low-level programs such as device drivers, but lacks the abstraction facilities of higher-level languages and demands careful manual tracking of resources such as memory buffers. As a result, many applications contain memory-handling bugs, which often manifest as serious security vulnerabilities. Researchers have done an admirable job of porting both Windows and Linux to a libOS model,[16] but for us this provided the perfect excuse to explore a less backward-compatible but more naturally integrated high-level language model. Figure 2 shows the logical workflow in MirageOS. Precise dependency tracking from source code (both local and global libraries) and configuration files lets the full provenance of the deployed kernel binaries be recorded in immutable data stores, sufficient to precisely recompile it on demand.

Stronger programming abstractions. High-level languages are steadily gaining ground in general application development and are increasingly used to glue components together via orchestration frameworks (for example, Puppet and Chef). Unfortunately, all this logic is typically scattered across software components and is written in several languages. As a result, it is difficult to reason statically about the whole system's behavior just by analyzing the source code.

Although OS virtualization has made the libOS possible without needing an army of device-driver writers, *protocol libraries* are still needed to replace the *services* of a traditional operating system.

MirageOS aims to unify these diverse interfaces—both kernel and application user spaces—into a single high-level language framework. Some of the benefits of modern programming languages include:

▸ *Static type checking.* Compilers can classify program variables and functions into types and reject code where a variable of one type is operated on as if it were a different type. Static type checking catches these errors at compile time rather than runtime and provides a flexible way for a systems programmer to protect different parts of a program from each other without depending solely on hardware mechanisms such as virtual memory paging. The most obvious benefit of type checking is the resulting lack of memory errors such as buffer or integer overflows, which are still prevalent in the CERT (Computer Emergency Readiness Team) vulnerability database. A more advanced use is capability-style access control,[19] which can be entirely enforced in a static type system such as ML's, as long as the code all runs within the same language runtime.

▸ *Automatic memory management.* Runtime systems relieve programmers of the burden of allocating and freeing memory, while still permitting manual management of buffers (for example, for efficient I/O). Modern garbage collectors are also designed to minimize application interruptions via incremental and generational collection, thus permitting their use in high-performance systems construction.[7,11]

▸ *Modules.* When the code base grows, modules partition it into logical components with well-defined interfaces gluing them together. Modules help software development scale as internal implementation details can be abstracted and the scope of a single source-code change can be restricted. Some module systems, such as those found in OCaml and Standard ML, are statically resolved at compilation time and are largely free of runtime costs. The goal is to harness these module systems to build whole systems, crossing traditional kernel and user-space boundaries in one program.

▸ *Metaprogramming.* If the runtime configuration of a system is partially understood at compile time, then a compiler can optimize the program

much more than it would normally be able to. Without knowledge of the runtime configuration, the compiler's hands are tied, as the output program must remain completely generic, just in case. The goal here is to unify configuration and code at compilation time and eliminate waste before deploying to the public cloud.

Together, these features significantly simplify the construction of large-scale systems: managed memory eliminates many resource leaks, type inference results in more succinct source code, static type checking verifies that code matches some abstraction criteria at compilation time rather than execution time, and module systems allow the manipulation of this code at the scales demanded by a full OS and application stack.

A Functional Prototype In OCaml

We started building the MirageOS prototype in 2008 with the intention of understanding how far we could unify the programming models underlying library operating systems and cloud-service deployment. The first design decision was to adopt the principles behind *functional programming* to construct the prototype. Functional programming has an emphasis on supporting abstractions that make it easier to track mutability in programs, and previous research has shown that this need not come at the price of performance.[11]

The challenge was to identify the correct modular abstractions to support the expression of an entire operating system and application software stack in a single manageable structure. MirageOS has since grown into a mature set of almost 100 open-source libraries that implement a wide array of functionality, and it is starting to be integrated into commercial products such as Citrix XenServer.[17]

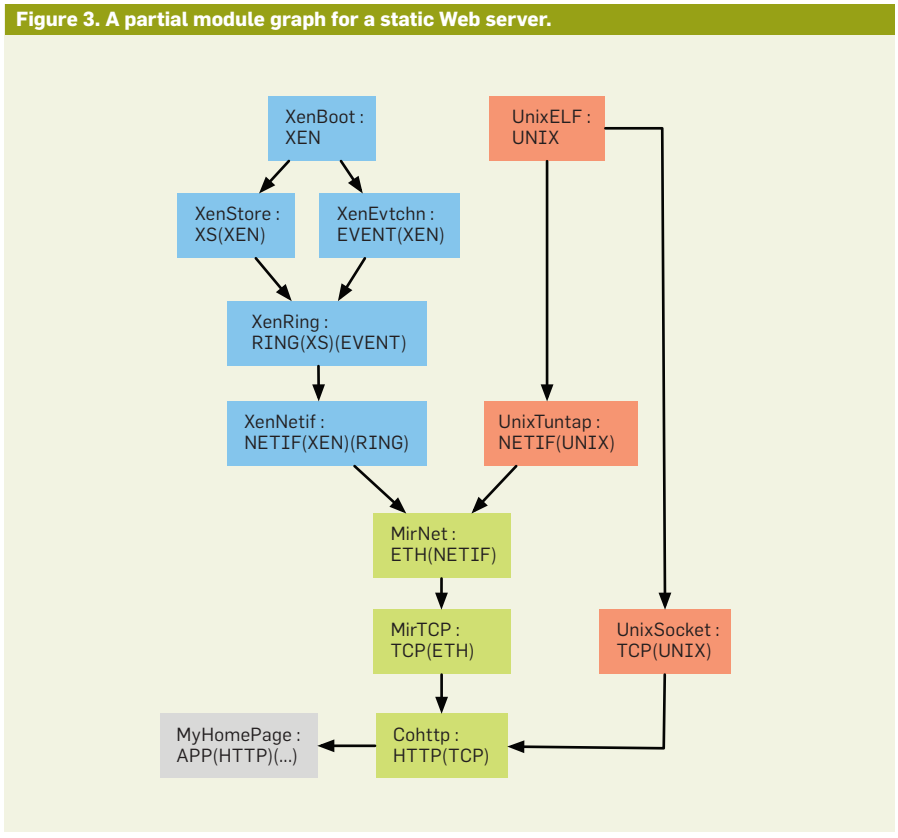
Figure 2 illustrates MirageOS's design. It grants the compiler a much broader view of source-code dependencies than a conventional cloud deployment cycle:

- All source-code dependencies of the input application are explicitly tracked, including all the libraries required to implement kernel functionality. MirageOS includes a build system that internally uses a SAT solver (using the OPAM package manager, with solvers from the Mancoosi project) to search for compatible module implementations from a published online package set. Any mismatches in interfaces are caught at compile time because of OCaml's static type checking.
- The compiler can then output a full stand-alone kernel instead of just a Unix executable. These unikernels are single-purpose libOS VMs that perform only the task defined in their application source and configuration files, and they depend on the hypervisor to provide resource multiplexing and isolation. Even the bootloader, which has to set up the virtual memory page tables and initialize the language runtime, is written as a simple library. Each application links to the specific set of libraries it needs and can glue them together in application-specific ways.
- The specialized unikernels are deployed on the public cloud. They have a significantly smaller attack surface than the conventional virtualized equivalents and are more resource efficient in terms of boot time, binary size, and runtime performance.

Why OCaml? OCaml is the sole base language for MirageOS for a few key reasons. It is a full-fledged systems programming language with a flexible programming model that supports functional, imperative, and object-oriented styles within a single, ML-inspired type system. It also features a portable single-threaded runtime that makes it ideal for porting to restricted environments such as a barebones Xen VM. The compiler heavily emphasizes static type checking, and the resulting binaries are fast native code with minimal runtime type information. Principal type inference allows type annotations to be safely omitted, and the module system is among the most powerful in a general-purpose programming language in terms of permitting flexible and safe code reuse and refactoring. Finally, there were several examples of large-scale uses of OCaml in industry[14] and within Xen itself,[17] and the positive results were encouraging before embarking on the large multiyear project that MirageOS turned out to be.

Modular operating-system libraries. OCaml supports the definition of *module signatures* (a collection of data-type and function declarations) that abstract the implementation of

Figure 3. A partial module graph for a static Web server.



module structures (definitions of concrete data types and functions). Modules can be parameterized over signatures, creating *functors* that define operations across other data types. (For more information about OCaml modules, functors, and objects, see *Real World OCaml*, published by O'Reilly and available at https://realworldocaml.org.) We applied the OCaml module system to breaking the usually monolithic OS kernel functionality into discrete units. This lets programmers build code that can be *progressively specialized* as it is being written, starting from a process in a familiar Unix environment and ending up with a specialized cloud unikernel running on Xen.

Consider a simple example. Figure 3 shows a partial module graph for a static Web server. Libraries are a module graph that abstract over operating-system functionality, and the OPAM package manager solves constraints over the target architecture. The application `MyHomePage` depends on a `HTTP` signature that is provided by the `Cohttp` library. Developers just starting out want to explore their code interactively using a Unix-style development environment. The `Cohttp` library needs a TCP implementation to satisfy its module signature, which can be provided by the `UnixSocket` library.

When the programmers are satisfied their HTTP logic is working, they can recompile to switch away from using Unix sockets to the OCaml TCP/IP stack shown by `MirTCP` in Figure 3. This still requires a Unix kernel but only as a shell to deliver Ethernet frames to the Web-server process (which now incorporates an OCaml TCP/IP stack as part of the application). The last compilation strategy drops the dependency on Unix entirely and recompiles the `MirNet` module to link directly to a Xen network driver, which in turn pulls in all the dependencies it needs to boot on Xen. This progressive recompilation is key to the usability of MirageOS, since we can evolve from the tried-and-tested Linux or FreeBSD functionality gradually but still end up with specialized unikernels that can be deployed on the public cloud. This modular operating-system structure has led to a number of other back ends being implemented in a similar vein to Xen. MirageOS now has experimental back ends that implement a simulator in NS3 (for large-scale functional testing), a FreeBSD kernel module back end, and even a JavaScript target by using the `js_of_ocaml` compiler. A natural consequence of this modularity is that it is easier to write portable code that defines exactly what it needs from a target platform, which is increasingly difficult on modern operating systems with the lack of a modern equivalent of Posix (which has led Linux, FreeBSD, Mac OS X, and Windows to have numerous incompatible APIs for high-performance services).

Figure 4. Virtual address space of the MirageOS Xen unikernel target.

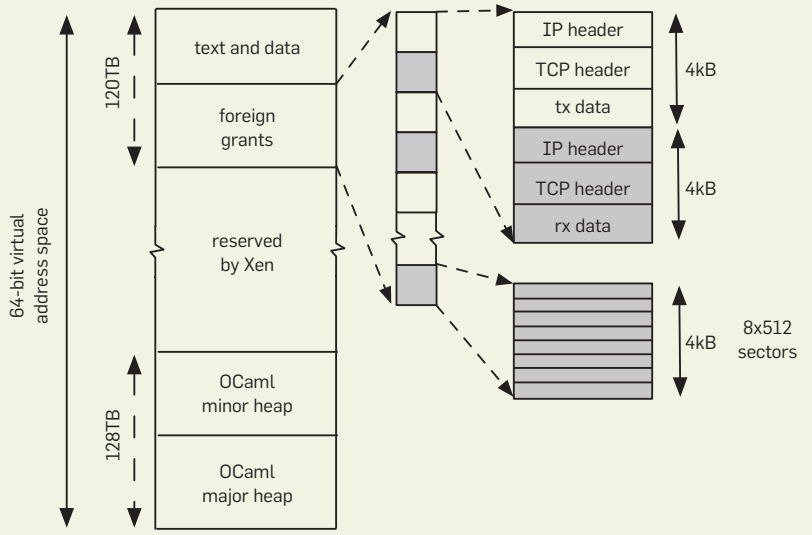
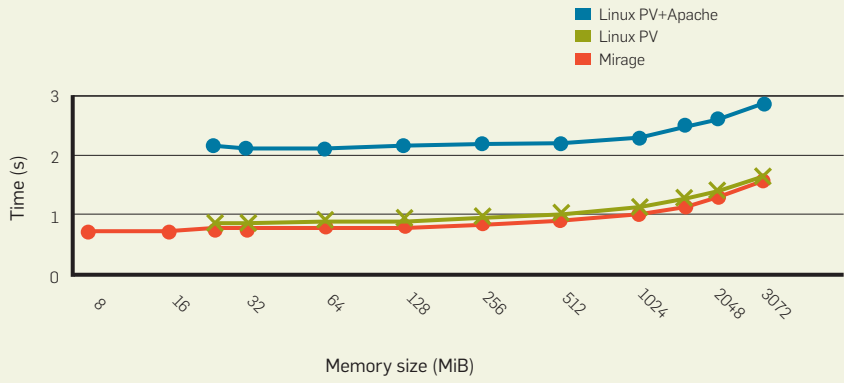


Figure 5. Boot time.



Configuration and state. Libraries in MirageOS are designed in as functional a style as possible: they are reentrant with explicit state handles, which are in turn serializable so that they can be reconstructed explicitly. An application consists of a set of libraries plus some configuration code, all linked together. The configuration is structured as a tree roughly like a file system, with subdirectories being parsed by each library to initialize their own values (reminiscent of the Plan 9 operating system). All of this is connected by *metaprogramming*—an OCaml program generates more OCaml code that is compiled until the desired target is reached.

The metaprogramming extends into storage as well. If an application

uses a small set of files (which would normally require all the baggage of block devices and a file system), MirageOS can convert it into a static OCaml module that satisfies the file-system module signature, relieving it of the need for an external storage dependency. The entire MirageOS home page (http://openmirage.org) is served in this manner.

One (deliberate) consequence of metaprogramming is that large blocks of functionality may be entirely missing from the output binary. This makes dynamic reconfiguration of the most specialized targets impossible, and a configuration change requires the unikernel to be relinked. The lines of active (that is, post-configuration) code involved in a MirageOS Web server are shown in Table 1, giving a sense of the small amount of code involved in such a recompilation.

Linking the Xen unikernel. In a conventional OS, application source code is first compiled into object files via a native-code compiler and then handed off to a linker that generates an executable binary. After compilation, a dynamic linker loads the executable and any shared libraries into a *process* with its own address space. The process can then communicate with the outside world by system calls, mediated by the operating-system kernel. Within the kernel, various subsystems such as the network stack or virtual memory system process system calls and interacts with the hardware.

In MirageOS, the OCaml compiler receives the source code for an entire kernel's worth of code and links it into a stand-alone native-code object file. It is linked against a minimal runtime that provides boot support and the garbage collector. There is no preemptive threading, and the kernel is event driven via an I/O loop that polls Xen devices.

The Xen unikernel compilation derives its performance benefit from the fact that the running kernel has a single virtual address space, designed to run only the OCaml runtime. The virtual address space of the MirageOS Xen unikernel target is shown in Figure 4. Since all configuration information is explicitly part of the compilation, there is no longer a need for the usual dynamic linking support that requires executable mappings to be added after the VM has booted.[13]

One downside to a unikernel is the burden it places on the cloud orchestration layers because of the need to schedule many more VMs with greater churn.

Benefits

Consider the life cycle of a traditional application. First the source code is compiled to a binary. Later, the binary is loaded into memory and an OS process is created to execute it. The first thing the running process will do is read its configuration file and specialize itself to the environment it finds itself in. Many different applications will run exactly the same binary, obtained from the same binary package, but with different configuration files. These configuration files are effectively additional program code, except they are normally written in ad hoc languages and interpreted at runtime rather than compiled.

Deployment and management. Configuration is a considerable overhead in managing the deployment of a large cloud-hosted service. The traditional split between the *compiled* (code) and *interpreted* (configuration) is unnecessary with unikernel compilation. Application configuration is code—perhaps as an embedded domain-specific language—and the compiler can analyze and optimize across the whole unikernel.

In MirageOS, rather than treating the database, Web server, and so on, as independent applications that must be connected by configuration files, they are treated as libraries within a single application, allowing the application developer to configure them using either simple library calls for dynamic parameters or metaprogramming tools for static parameters. This has the useful effect of making configuration decisions explicit and programmable in a host language rather than manipulating many ad hoc text files and thus benefiting from static-analysis tools and the compiler's type checker. The result is a big reduction in the effort needed to configure complex multiservice application VMs.

One downside to a unikernel is the burden it places on the cloud orchestration layers because of the need to schedule many more VMs with greater churn (since every reconfiguration requires the VM to be redeployed). The popular orchestration implementations have

grown rather organically in recent years and consist of many distributed components that are not only difficult to manage, but also relatively high in latency and resource consumption.

One of the first production uses for MirageOS is to fix the cloud-management stacks by evolving the OCaml code within XenServer[17] toward the structured unikernel worldview. This turns the monolithic management layer into a more agile set of intercommunicating VMs that can be scheduled and restarted independently. MirageOS makes constructing these single-purpose VMs easy: they are first built and tested as regular Unix applications before flipping a switch and relinking against the Xen kernel libraries (http://openmirage.org/blog/xenstore-stub-domain). When they are combined with Xen driver domains,[3] they can dramatically increase the security and robustness of the cloud-management stack.

Resource efficiency and customization. The cloud is an environment where all resource usage is metered and rented. At the same time, multitenant services suffer from variability in load that encourages rapid scaling of deployments—both *up* to meet current demand and *down* to avoid wasting money. In MirageOS, features that are not used in a particular build are not included, and whole-system optimization techniques can be used to eliminate waste at compilation time rather than deployment time. In the most specialized mode, all configuration files are statically evaluated, enabling extensive dead-code elimination at the cost of having to recompile to reconfigure the service.

The small binary size of the unikernels (on the order of hundreds of kilobytes in many cases) makes deployment to remote data centers across the Internet much smoother. Boot time is also easily less than a second, making it feasible to boot a unikernel in response to incoming network packets.

Figure 5 shows the comparison between the boot time of a service in MirageOS and a Linux/Apache distribution. The boot time of a stripped-down Linux kernel and MirageOS are similar, but the inefficiency creeps into Linux as soon as it has to initialize the user-space applications. The MirageOS unikernel is ready to serve traffic as soon as it boots.

The MLton[20] compiler pioneered WPO (whole program optimization), where an application and all of its libraries are optimized together. In the libOS world, a whole program is actually a whole operating system: this technique can now optimize all the way from application-level code to low-level device drivers. Traditional systems eschew WPO in favor of dynamic linking, sometimes in combination with JIT (just-in-time) compiling, where a program is analyzed dynamically, and optimized code is generated on the fly. Whole-program, compile-time optimization is more appropriate for cloud applications that care about resource efficiency and reducing their attack surface. Other research elaborates on the security benefits.[13]

An interesting recent trend is a move toward operating-system *containers* in which each container is managed by the same operating-system kernel but with an isolated file system, network, and process group. Containers are quick to create since there is no need to boot a new kernel, and they are fully compatible with existing kernel interfaces. However, these gains are made at the cost of reduced security and isolation; unikernels share only the minimal hypervisor services via a small API, which is easy to understand and audit. Unikernels demonstrate that layering language runtimes onto a hypervisor is a viable alternative to lightweight containers.

A new frontier of portability. The structure of MirageOS libraries shown in Figure 3 explicitly encodes what the library needs from its execution environment. While this has conventionally meant a Posix-like kernel and user space, it is now possible to compile OCaml into more foreign environments, including FreeBSD kernel modules, JavaScript running in the browser, or (as the Scala language does) directly targeting the Java Virtual Machine (JVM).

Some care is still required for execution properties that are not abstractable in the OCaml type system. For example, floating-point numbers

Table 1. Approximate size of libraries used by a typical MirageOS unikernel running a Web server.

Library	C/kLOC	OCaml/kLOC
Boot	18	0
OCaml runtime	20	0
threads	5	27
interdomain comms	trace	1
network driver	0	1
TCP/IP	trace	12
block driver	0	1
HTTP	0	11
Total	**43**	**52**

Table 2. Other unikernel implementations.

Unikernel	Language	Targets
Mirage[13]	OCaml	Xen, kFreeBSD, POSIX, WWW/js
Drawbridge[17]	C	Windows "picoprocess"
HalVM[8]	Haskell	Xen
ErlangOnXen	Erlang	Xen
OSv[2]	C/Java	Xen, KVM
GUK	Java	Xen
NetBSD "rump"[9]	C	Xen, Linux kernel, POSIX
ClickOS[14]	C++	Xen

are generally forbidden when running as a kernel module; thus, a modified compiler emits a type error if floating-point code is used when compiling for that hardware target.

Other third-party OCaml code often exhibits a similar structure, making it much easier to work under MirageOS. For example, Arakoon (http://arakoon.org) is a distributed key-value store that implements an efficient multi-Paxos consensus algorithm. The source-code patch to compile it under MirageOS touched just two files and was restricted to adding a new module definition that mapped the Arakoon back-end storage to the Xen block driver interface.

Unikernels in the Wild

MirageOS is certainly not the only unikernel that has emerged in the past few years, although it is perhaps the most extreme in terms of exploring the clean-slate design space. Table 2 shows some of the other systems that build unikernels. HalVM[8] is the closest to the MirageOS philosophy, but it is based on the famously pure and lazy Haskell language rather than the strictly evaluated OCaml. On the other end of the spectrum, OSv[2] and rump kernels[9] provide a compatibility layer for existing applications, and deemphasize the programming model improvements and type safety that guides MirageOS. The Drawbridge project[16] converts Windows into a libOS with just a reported 16MB overhead per application, but it exposes higher-level interfaces than Xen (such as threads and I/O streams) to gain this efficiency.

Ultimately, the public cloud should support all these emerging projects as first-class citizens just as Linux and Windows are today. The Xen Project aims to support a brave new world of *dust clouds*: tiny one-shot VMs that run on hypervisors with far greater density than is currently possible and that self-scale their resource needs by constantly calling into the cloud fabric. The libOS principles underlying MirageOS mean it is not limited to running on a hypervisor platform—many of the libraries can be compiled to multiscale environments,[12] ranging from ARM smartphones to bare-metal kernel modules. To understand the implications of this flexibility, we have been exploring use cases ranging from managing personal data[4] and facilitating anonymous communication,[15] to building software-defined data-center infrastructure.

Acknowledgments

The MirageOS effort has been a large one and would not be possible without the intellectual and financial support of several sources. The core team of Richard Mortier, Thomas Gazagnaire, Jonatham Ludlam, Haris Rotsos, Balraj Singh, and Vincent Bernardoff have toiled to help us build the clean-slate OCaml code, with constant support and feedback from Jon Crowcroft, Steven Hand, Ian Leslie, Derek McAuley, Yaron Minsky, Andrew Moore, Simon Moore, Alan Mycroft, Peter G. Neumann, and Robert N.M. Watson. Space prevents us from fully acknowledging all those who contributed to this effort. We encourage readers to visit http://queue.acm.org for our full list.

This work was primarily supported by Horizon Digital Economy Research, RCUK grant EP/G065802/1. A portion was sponsored by DARPA (Defense Advanced Research Projects Agency) and AFRL (Air Force Research Laboratory), under contract FA8750-11-C-0249. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of DARPA or the Department of Defense.

MirageOS is available freely at http://openmirage.org. We welcome feedback, patches, and improbable stunts using it.

Related articles on queue.acm.org

Self-Healing in Modern Operating Systems
Michael W. Shapiro
http://queue.acm.org/detail.cfm?id=1039537

Erlang for Concurrent Programming
Jim Larson
http://queue.acm.org/detail.cfm?id=1454463

Passing a Language through the Eye of a Needle
Roberto Ierusalimschy, Luiz Henrique de Figueiredo and Waldemar Celes
http://queue.acm.org/detail.cfm?id=1983083

OCaml for the Masses
Yaron Minsky
http://queue.acm.org/detail.cfm?id=2038036

References

1. Barham, P. et al. Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles* (2003), 164–177.
2. Cloudius Systems. OSv; https://github.com/cloudius-systems/osv.
3. Colp, P. et al. A. Breaking up is hard to do: Security and functionality in a commodity hypervisor. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles* (2011), 189–202.
4. Crowcroft, J., Madhavapeddy, A., Schwarzkopf, M., Hong, T. and Mortier, R. Unclouded vision. In *Proceedings of the International Conference on Distributed Computing and Networking*, 29–40.
5. Eisenstadt, M. My hairiest bug war stories. *Commun. ACM 40*, 4 (Apr. 1997), 30–37.
6. Engler, D. R., Kaashoek, M. F. and O'Toole, Jr., J. Exokernel: An operating system architecture for application-level resource management. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, (1995), 251–266.
7. Eriksen, M. Your server as a function. In *Proceedings of the 7th Workshop on Programming Languages and Operating Systems*, (2013), 5:1–5:7.
8. Galois Inc. The Haskell Lightweight Virtual Machine (HaLVM) source archive; https://github.com/GaloisInc/HaLVM.
9. Kantee, A. Flexible operating system internals: The design and implementation of the anykernel and rump kernels. Ph.D. thesis, Aalto University, Espoo, Finland, 2012.
10. Leslie, I.M. et al. The design and implementation of an operating system to support distributed multimedia applications. *IEEE Journal of Selected Areas in Communications 14*, 7 (1996), 1280–1297.
11. Madhavapeddy, A., Ho, A., Deegan, T., Scott, D. and Sohan, R. Melange: Creating a "functional" Internet. *SIGOPS Operating Systems Review 41*, 3 (2007), 101–114.
12. Madhavapeddy, A., Mortier, R., Crowcroft, J. and Hand, S. Multiscale not multicore: Efficient heterogeneous cloud computing. In *Proceedings of ACM-BCS Visions of Computer Science. Electronic Workshops in Computing*, (Edinburgh, U.K., 2010).
13. Madhavapeddy, A. et al. Unikernels: Library operating systems for the cloud. In *Proceedings of the 18th International Conference on Architectural Support for Programming Languages and Operating Systems*, (2013), 461–472.
14. Minsky, Y. OCaml for the masses. *Commun. ACM 54*, 11 (Nov. 2011), 53–58.
15. Mortier, R., Madhavapeddy, A., Hong, T., Murray, D. and Schwarzkopf, M. Using dust clouds to enhance anonymous communication. In *Proceedings of the 18th International Workshop on Security Protocols* (2010).
16. Porter, D.E., Boyd-Wickizer, S., Howell, J., Olinsky, R. and Hunt, G.C. Rethinking the library OS from the top down. In *Proceedings of the 16th International Conference on Architectural Support for Programming Languages and Operating Systems*, (2011), 291–304.
17. Scott, D., Sharp, R., Gazagnaire, T. and Madhavapeddy, A. Using functional programming within an industrial product group: perspectives and perceptions. In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, (2010), 87–92.
18. Vinge, V. *A Fire Upon the Deep*. Tor Books, New York, NY, 1992.
19. Watson, R.N.M. A decade of OS access-control extensibility. *Commun. ACM 56*, 2 (Feb. 2013), 52–63.
20. Weeks, S. Whole-program compilation in MLton. In *Proceedings of the 2006 Workshop on ML*.

Anil Madhavapeddy is a Senior Research Fellow at the University of Cambridge, based in the Systems Research Group. He was on the original team that developed the Xen hypervisor and XenServer management toolstack written in OCaml. XenServer has been deployed on millions of hosts and drives critical infrastructure for many Fortune 500 companies.

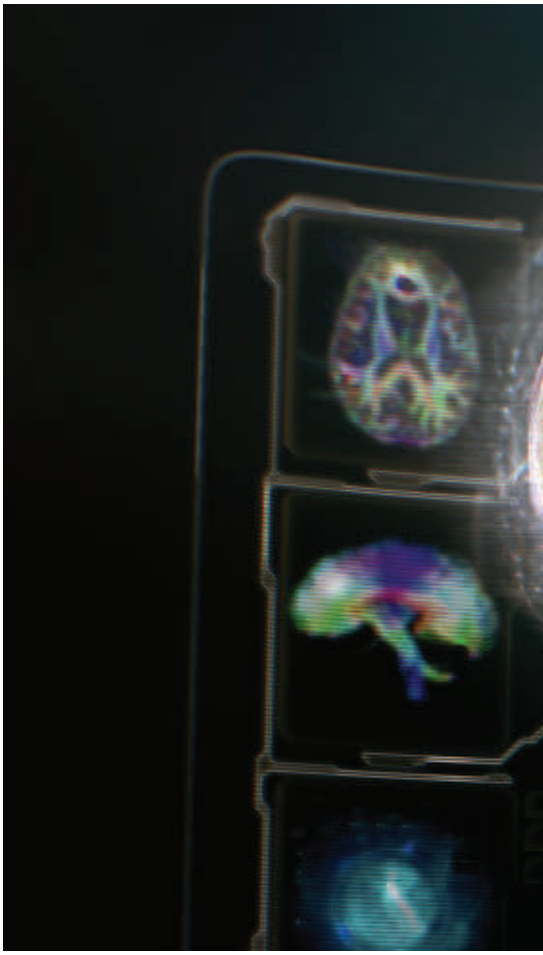
Dave Scott is a Principal Architect at Citrix Systems where he works on the XenServer virtualization platform. His focus is on improving XenServer reliability and performance through exploiting advances in open-source software and high-level languages.

DOI:10.1145/2541883.2541899

Touchless interaction with medical images lets surgeons maintain sterility during surgical procedures.

BY KENTON O'HARA, GERARDO GONZALEZ, ABIGAIL SELLEN, GRAEME PENNEY, ANDREAS VARNAVAS, HELENA MENTIS, ANTONIO CRIMINISI, ROBERT CORISH, MARK ROUNCEFIELD, NEVILLE DASTUR, AND TOM CARRELL

Touchless Interaction in Surgery

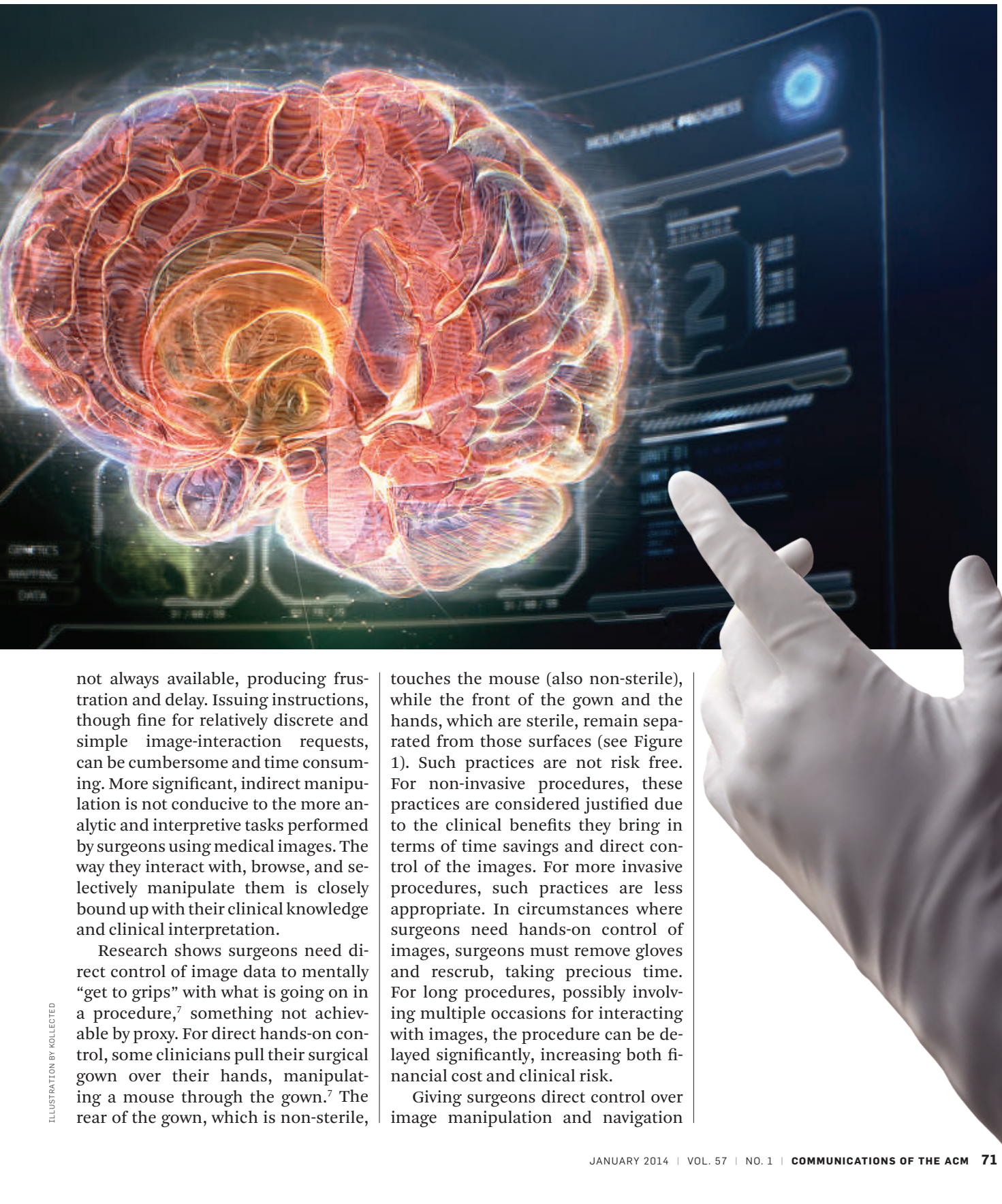


A GLANCE AROUND any operating theatre reveals many visual displays for accessing pre- and intra-operative images, including computer tomography (CT), magnetic resonance imagery (MRI), and fluoroscopy, along with various procedure-specific imaging applications. They support diagnosis and planning and provide a virtual "line of sight" into the body during surgery. Although surgeons rely on the capture, browsing, and manipulation of these images, they are constrained by typical interaction mechanisms (such as keyboard and mouse).

At the heart of the constraints is the need to maintain a strict boundary between what is sterile and what is not. When surgeons are scrubbed and gloved, they cannot touch these input devices without breaking asepsis. To get around it, several strategies are available for interacting with images, though they are often not ideal; for example, surgeons commonly request other members of the surgical team (such as radiographers and nurses) to manipulate images under their instruction.[7,11] While it can succeed, it, too, is not without complications. Team members are

» key insights

- **Beyond demonstrating technical feasibility, touchless interaction in surgery should be designed to work within operating-theatre practices.**
- **Gesture design should consider not only individual interaction with medical images but how they are used in the context of collaborative discussion.**
- **Gesture design across one and two hands should accommodate expressive richness, as well as the surgeon's hands, but is constrained by the close proximity of the surgical team and movement restrictions due to sterile practice.**



not always available, producing frustration and delay. Issuing instructions, though fine for relatively discrete and simple image-interaction requests, can be cumbersome and time consuming. More significant, indirect manipulation is not conducive to the more analytic and interpretive tasks performed by surgeons using medical images. The way they interact with, browse, and selectively manipulate them is closely bound up with their clinical knowledge and clinical interpretation.

Research shows surgeons need direct control of image data to mentally "get to grips" with what is going on in a procedure,[7] something not achievable by proxy. For direct hands-on control, some clinicians pull their surgical gown over their hands, manipulating a mouse through the gown.[7] The rear of the gown, which is non-sterile, touches the mouse (also non-sterile), while the front of the gown and the hands, which are sterile, remain separated from those surfaces (see Figure 1). Such practices are not risk free. For non-invasive procedures, these practices are considered justified due to the clinical benefits they bring in terms of time savings and direct control of the images. For more invasive procedures, such practices are less appropriate. In circumstances where surgeons need hands-on control of images, surgeons must remove gloves and rescrub, taking precious time. For long procedures, possibly involving multiple occasions for interacting with images, the procedure can be delayed significantly, increasing both financial cost and clinical risk.

Giving surgeons direct control over image manipulation and navigation

ILLUSTRATION BY KOLLECTED

while maintaining sterility within the operating theatre is a key goal,[20] one that has captured the imagination of research groups and commercial entities worldwide. For some, the approach is to insert a barrier between the sterile gloves of the surgeon and a nonsterile interaction device (such as IDEO's optical mouse-in-a-bag solution[5]). While such solutions reflect a certain elegance in their simplicity, there remain certain practical concerns at the patient bedside. In addition, barrier-based solutions involve certain inherent risks due to the potential for damage to the barrier. Other approaches have sought to enable interaction techniques in the operating theatre that avoid the need for contact with an input device altogether. The seeds of this interest were in evidence in the mid-2000s when computer-vision techniques were first used for controlling medical-imaging systems by tracking the in-air gestures of the surgeon. Graetzel et al.,[4] in an early example of touchless medical imaging, let surgeons control standard mouse functions (such as cursor movement and clicking) through camera-tracked hand gestures. Shortly afterward, more sophisticated air-based gestures were used for surgical-imaging technology in the form of Wachs et al.'s Gestix system.[21] Rather than just emulate mouse functionality, Gestix introduced possibilities for more bespoke gesture-based control (such as for navigation, zooming, and rotation).

These initial systems paved an important path, and, more recently, the number of systems and research efforts considering touchless control of medical images for surgical settings has grown significantly, as covered by Ebert et al.,[1,2] Gallo,[3] Johnson et al.,[7] Kipshagen et al.,[9] Mentis et al.,[11] Mithun et al.,[13] O'Hara et al.,[15] Ruppert et al.,[16] Stern et al.,[17] Strickland et al.,[18] and Tan et al.[19] One enabler of this growth is the Kinect sensor and software development kit,[12] which has lowered barriers to entry, including financial cost, development complexity, and need to wear trackable markers. The Kinect sensor is based on a laser and horizontally displaced infrared (IR) camera. The laser projects a known pattern onto the scene. The depth of each point in the scene is estimated by analyzing the way the pattern deforms when viewed from the Kinect's IR camera.

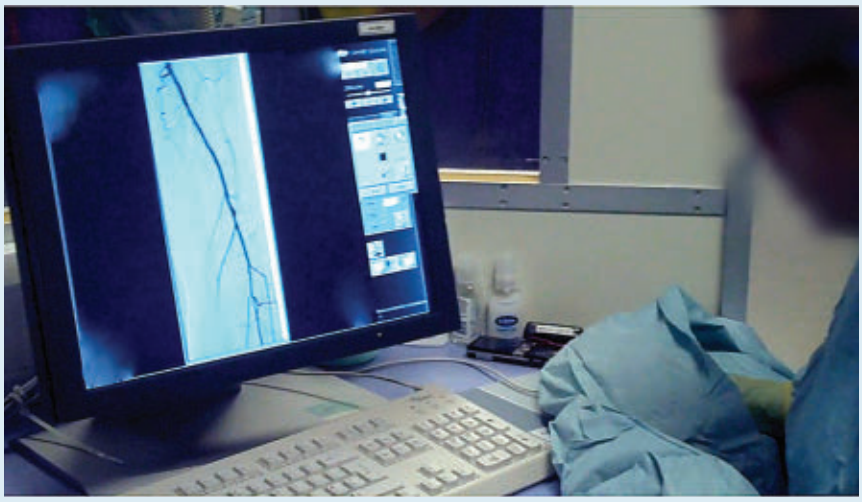
When scene depth is estimated, a machine-learning-based algorithm automatically interprets each pixel as belonging to the background or to one of the 31 parts in which the controlling person's body has been subdivided. This information is then used to compute the position of the "skeleton," a stickman representation of the human controller. Kinect has helped overcome some of the inherent challenges of full-depth skeleton capture with purely camera-based systems. With this range of systems, common themes have emerged, along with opportunity to explore a more diverse set of approaches to this particular problem area of touchless interaction during surgery. The concern is no longer to demonstrate the technical feasibility of such solutions but how to best design and implement touchless systems to work within the particular demands and circumstances that characterize practices in the operating theatre. Reflecting them, and with the growing interest in the technology, we highlight some lessons learned, as well as issues and challenges relevant to the development of these systems, beginning with key projects.

A leading example involves the system used for multiple kinds of surgery at Sunnybrook Hospital in Toronto[18] in which a Kinect helps navigate a predefined stack of MRI or CT images, using a simple constrained gesture set to move forward or backward through the images and engage and disengage from the system, an important issue revisited later.

Any image transformation (such as rotating, zooming, or other parameter adjustments) is not available in the system unless these manipulations are integrated into the predefined image stack. The simplicity of the Sunnybrook system reflects a genuine elegance. The limited number of gestures yields benefits in terms of ease of use and system learnability. Such a constrained-gesture set can also offer certain reliability benefits: enabling use of reliably distinctive gestures while avoiding "gesture bleed," where gestures in a vocabulary share common kinaesthetic components, possibly leading to system misinterpretation. Given that the system is one of only a few actively deployed and in use today, such reliability concerns are paramount in the design choices made by its developers. Note, too, adoption of two-handed gestures in the design of the gestural vocabulary, a technique that can yield certain benefits, as well as constrain the way the system is used in surgical contexts, a key theme covered later.

While the Sunnybrook system reflects elegant simplicity, such an approach must also address inherent limitations. Interaction with medical images in surgical settings often extends beyond simple navigation, requiring a much richer set of image-manipulation options beyond rotate/pan/zoom to potentially include ad-

Figure 1. Surgical gown used to avoid touching non-sterile mouse with sterile gloved hand.



justment of various image parameters (such as density functions to reveal features like bone, tissue, and blood vessels, and opacity). Such possibilities may even include marking up or annotating images during procedures. Moreover, manipulation may apply to whole images or more specific regions of interest defined by the clinician. With these possibilities in mind, several recent projects involving Kinect-based touchless interaction with surgical images have developed a much larger gesture set to accommodate the increased functionality, as well as to interface with standardized open source Digital Imaging and Communications in Medicine (DICOM) image viewers and Picture Archive and Communication System (PACS) systems (such as Medical Imaging TOolkit and OsiriX). Notable examples are systems developed by Ebert et al.,[1,2] Gallo et al.,[3] Ruppert et al.,[16] and Tan et al.[19]

Incorporating these richer functional sets is impressive but involves notable challenges. One concerns the notion of expressive richness, or how to map an increasingly large set of functionalities (often involving the continuous adjustment of levels of a parameter) onto a reliably distinctive gesture vocabulary. Several approaches have been applied in these systems (such as use of modes to distinguish gestures and input modalities, including speech and composite multi-handed gestures). For example, using one- and two-handed tracking not only yields the benefits of bi-manual interaction but enables a richer set of expressive possibilities. In both Ebert et al.[1,2] and Gallo et al.,[3] the gesture set employs both one- and two-handed gestures. Different gesture combinations (such as single-hand, two hands together, and two hands apart) can then be used to denote particular image parameters that can be adjusted according to their respective positioning in the *x*, *y*, and *z* planes. More recent versions of the Ebert et al.[1] system include further expressive capabilities through algorithms that recognize more finger-level tracking in which spread hands are distinguishable from, say, open-palm hands.

Along with the larger gesture sets enabled by this expressiveness comes concern over the learnability of the systems,[14] particularly as new system functionalities may have to be accommodated. We see attempts to deal with these issues in the systems of Ruppert et al.[16] and Tan et al.,[19] building up compound gestures that combine dominant and non-dominant hands in a consistent and extendable way. The non-dominant hand is used for selecting particular functions or modes, while the dominant hand moves within the *x*, *y*, and *z* planes, for the continuous adjustment of image parameters. In this way common gestures can be applied across a range of functionalities, making the system more learnable and extendable.

It is not so much that more than one person wants to control images simultaneously but that sometimes one person must be able to fluidly hand over control to another person.

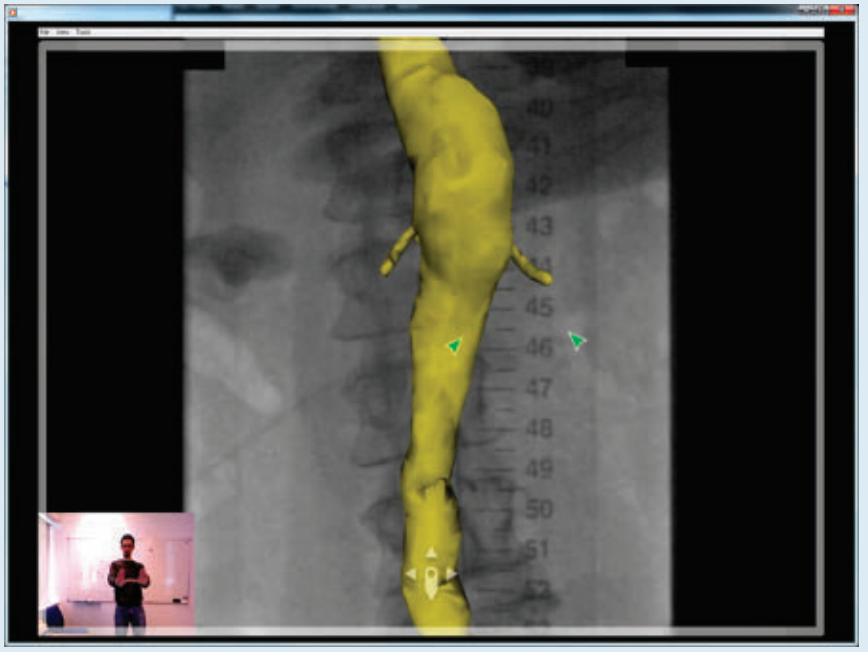
What emerges is the use of one- and two-handed gestures as an important theme in the design and understanding of touchless medical systems we pick up again later. In particular, while the varied approaches appear to be motivated by certain control pragmatics (such as need for expressive richness and learnability), what is not apparent is how particular design decisions are motivated by principles of bimanual interaction design[10] or more significantly the broader set of socio-technical issues that arise when considering how these systems might be used in the context of an actual surgical procedure.

Another possibility is the use of voice recognition, as seen in the work of Ebart et al.[2] and in our own work.[15] However, voice-recognition software involves special challenges in noisy operating theatres so, when used in isolation, may not be suitable for manipulation of continuous parameters. But most significant in the use of voice in these systems is how it can be combined with gestural modality to achieve control; for discrete actions and functions (such as changing mode and functionality) voice control could deliver important benefits.

Socio-technical Concerns

A central concern goes beyond simply developing touchless control mechanisms to overcome sterility requirements. First, they need to be situated in the context of working practices performed by the surgical team and in the setting of an operating theatre. Such settings and practices shape and constrain system design choices

Figure 2. Gesture system for manipulating 3D overlay in vascular surgery.



involving, say, tracking algorithms, gesture vocabulary, and distribution of interaction across different input modalities, including voice and gesture. While many of the systems discussed here were developed in collaboration with and successfully used by clinical partners, the rationale behind their design choices often remains implicit with respect to the settings and work practices in which they are deployed. As the field grows, it is worth reflecting on these issues and making them more explicit. To do this we draw on our experience developing a system for use in vascular surgery and how its design choices relate to particular socio-technical concerns following observations in the operating theatre. The focus on our own experience is for illustrative purposes, our intention being to highlight lessons for the broader set of technologies we discuss.

The system we developed was for use during image-guided vascular surgery at Guy's and St. Thomas' Hospital (GSTT) in London, U.K. During such a procedure, the surgeon is continuously guided by live fluoroscopy and x-ray images on a bank of monitors above the patient table. On one of them, a volumetric rendering of the aorta (from preoperative CT data) is overlaid on continuously updated x-ray images to help the surgeon visualize where the inserted wires and stents are located with respect to the actual structure of the aorta. This combined overlay is manipulated through the system's Kinect-based gesture and voice recognition (see Figure 2 and Figure 3).

In designing the system we had to address notable socio-technical concerns with broader significance for how to think about system development, including collaboration and control, engagement and disengagement, and image inspection with one hand, two hands, and hands-free.

Collaboration and control. In many systems, the design focus is on providing a single point of control for the surgeon in the operating theatre. While this remains an important goal, surgery involves significant collaborative aspects of imaging practices (such as in Johnson et al.[7] and Mentis et al.[11]). It is not so much that more than one person wants to control images simultaneously but that sometimes one person must be able to fluidly hand over control to another person; for example, if the surgeon is busy with the procedure and patient management, other clinical support may have to assume control of the images. Other times, the clinician leading the procedure might hand over certain responsibilities to a specialist or trainee. A second significant collaborative issue concerns collaborative clinical interpretation and discussion in which the various members of the surgical team point and gesticulate around the displayed images.

In the GSTT system we sought to address them by tracking the skeletons of multiple team members, using color-coding to give them a distinct pair of cursors corresponding to their hands. This color-coding of cursors allows collaborators to point and gesticulate at different parts of the image as they discuss, interpret, and plan an appropriate course of action. At any point, they can raise their hands and issue a spoken command to request control of the system so, as with the other systems covered here, there is a notion of a single dominant controller of the images. However, even in this mode, other team members can point and gesture through visible cursors, assuming control at any time through voice command, if required by the procedure.

System engagement, disengagement. Gesturing before a screen is not always for the purpose of system control. Along with gesture in support of conversation, movement before a screen may result from other actions performed in the context of the procedure or as the surgeon attempts to transition between gestures. These actions raise the possibility of the system inadvertently recognizing them as system-control gestures. Key in the design of the systems then is the need for mechanisms to move between states of system and engagement and disengagement, reinforced with appropriate feedback to signal the system state.

Multiple approaches are seen in various systems, each bringing its own set of pros and cons; for example, in the Sunnybrook system,[18] the developers incorporated a deliberately unusual gesture above the head to engage/disengage the system. Such a gesture is not likely to occur in the course of other activity so can be considered useful in terms of avoiding inadvertent triggering. In developing our system, we tried a number of approaches with varying success; for example, to engage the system to recognize gestures, we initially used a right-handed "waving"

gesture that suffered from "gesture transition," whereby the movement necessary to initiate the hand-wave gesture was sometimes recognized as a discrete gesture in and of itself.

This misinterpretation relates to the notion of "gesture spotting," or detection of the start and end points of a gesture through low-level kinaesthetic features (such as acceleration[8]). While gesture-spotting techniques are improving, it remains an inherently difficult challenge for system developers. One way to address it is through non-classification-based techniques, whereby continuous image parameters correspond to continuous positioning of the controller's hands. But such approaches are prone to the gesture-transition problem for a multiple reasons (such as when parameter adjustment extends beyond the reach of natural arm movements in either plane and particular areas of the screen are used for additional feature access). To address them, we incorporated a clutching mechanism in which arms are withdrawn close to the body to declutch the system, allowing movement transition without corresponding image manipulation by the system.

We also adopted a time-based lock in which surgeons hold their hands in position for a few seconds. While successful in other domains of gestural interaction, our evaluations with surgeons found a natural tendency for them to pause and inspect the image or hold a pose to point at a specific feature in the image. These behaviors clashed with the pause-based lock gesture, leading us to modify the system so engaging and disengaging control is achieved through a simple voice command that complements the gesture vocabulary and works well when a discrete change of state is needed.

Other developers have also explored automatic determination of intention to engage and disengage from a system; a good example is the work of Mithun et al.,[13] discussing contextual cues (such as gaze, hand position, head orientation, and torso orientation) to judge whether or not a surgeon intends to perform a system-readable gesture. Such approaches show promise in avoiding unintentional gestures, though determining human intent on the basis of these cues remains a challenge; for example, contextual cues are likely to be similar when talking and gesticulating around the image during collaborative discussion (such as when intending to interact with the system).

One hand, two hands, hands-free. Some systems discussed here make use of both one- and two-handed gestures. Besides increasing the richness of gesture vocabulary and exploiting important properties of bimanual action during interaction, important clinical considerations are at play in the ways systems are designed for one or two hands; for example image interaction is sometimes needed when a surgeon holds certain medical instruments, raising questions as to how many hands are available to perform certain gestural operations at a particular moment. The design of the gestural vocabulary is not just a question of having the right number of commands to match functionality but also how to reflect the clinical context of use as well.

Our GSTT system uses a range of one- and two-handed gestures. For panning and zooming an image, our observations and interviews with surgeons suggested these manipulations are typically done at points when instruments and catheter wires can be put down. For fading the opacity of the overlay and annotating the overlay with markers (such as to highlight a point of correspondence on the underlying fluoroscopy image), the surgeon may be holding onto the catheter, thereby leaving only one hand free. For these clinical reasons the system uses two-handed gestures for panning and zooming, but for opacity fading, the gesture can be performed with the hand that is free. For marking the image overlays the system combines one-handed tracking with a voice command, allowing the command to be carried out while holding the catheter.

This is not to say touchless control should be available at all times clinicians are using other instruments. There are indeed many points in a procedure when image manipulation could be a distraction to the task at hand. But there are opportunities for a combination to be considered by system developers; as a consequence, the specification of gesture vocabulary across both hands must be defined with clinical significance. Different kinds of surgical procedures clearly involve different constraints in terms of how and when image-manipulation opportunities can be combined with the use of surgical instruments, thus calling for careful

Figure 3. Gesture system for vascular surgery in theatre.

Figure 4. Interacting with medical images: (a) away from the operating table; and (b) at the operating table.

(a)

(b)

consideration of how to accomplish input, especially when both hands may be holding instruments. In such circumstances it might be possible to exploit voice commands for hands-free manipulation (providing they are suitable to the discrete properties of voice commands) or combine voice with other input (such as foot pedals, gaze input, and head movement). The point is when designing these systems developers must take an approach based not simply on technical but also on clinical demands as to whether one, two, or no hands are free for image interaction.

At the operating table, away from the operating table. This points to another important consideration—the physical location of surgeons when they need to interact with different imaging systems (see Figure 4). Aside from the use of tools at the operating table, the operating table can be a crowded environment, with surgeons often in close proximity to other members of the surgical team. Not only can this affect a system's approach to tracking but also impose constraints on the kinds of movement available for gesture design (such as physical restrictions of working in close proximity to others and those due to strict sterile practice). In sterile practice, hand movements must be restricted to the area extending forward from the surgeon's torso between the hips and chest level; shoulders, upper chest, thighs, and back are considered greater risks for compromising sterility, so movements in these areas (and thereby gestures) must be avoided. Moreover, the operating table itself hides the lower half of the surgeon's body, while away from the table the surgeon reveals more of the whole body to the tracking system. Kinect offers two tracking modes: default, optimized for full-body-skeleton tracking, and seating, optimized for upper-torso tracking (head, shoulders, and arms). While full-body tracking suits the situation in Figure 4a, the upper torso-tracking mode is better suited for the one in Figure 4b.

The surgeon's position at the operating table is defined by the clinical demands of the procedure so is not always in an ideal position in front of the gesture-sensing equipment in terms of distance and orientation. System developers may have to account for and accommodate such variations in the design of gestures and tracking capabilities. The examples here are intended to illustrate the broader issues, though developers may want to consider other clinically dependent and theatre-dependent configurations (such as a surgeon sitting in front of a PACS, system, as in Figure 1).

Conclusion

The goal is not simply to demonstrate the feasibility of touchless control in clinical settings; important design challenges range from the gesture vocabulary to the appropriate combination of input modalities and specific sensing mechanisms. We have shown how they can play out in the development of the systems but must be addressed further, especially as used in real-world clinical settings. This is not a straightforward matter of requesting clinicians specify what they want by way of a gesture vocabulary.

While clinician participation in the design process is essential it is not just a matter of offloading gesture design to clinicians. Rather, it is about system developers understanding how the work of clinical teams is organized with respect to the demands of the procedure and the particular properties of the physical setting

(such as a clinical team's positioning and movement around the patient, colleagues, and artifacts).

Developers must also view these systems not simply as sterile ways to perform the same imaging as they would without them but must understand what clinicians are trying to achieve through their imaging practices and how they are shaped by features of the procedure with respect to sterility. Combining this principle with an understanding of the technical properties of touchless systems, system developers can then drive design with a view to how they enable image interpretation, communication, and coordination among clinical team members.

Related is the need to evaluate the system as it will be used in the real world. The concern here is less basic usability than how it could change the practice of the surgical team, what it needs to do to accommodate the team, and the factors that constrain the way the system is used. One important consideration here is fatigue, or "gorilla arm," due to prolonged use in theatre that could affect system use, as well as other physical features of surgical practice.

While our focus here is overcoming the constraints of sterility in the operating theatre, a much broader issue involves infection control in hospital settings involving multiple devices, systems, and applications—from large displays to mobile units like tablet computers—for which touchless interaction mechanisms could play a role not just for medical professionals but for patients as well. Interesting examples include the GestureNurse system[6] in which a robotic surgical assistant is controlled through gesture-based commands.

Consider, too, 3D imaging in the operating theatre. Interpreting the enormous number of images produced by scanning technologies is cumbersome through traditional slice-by-slice visualization-and-review techniques. With the volumetric acquisition of scans, the data is increasingly visualized as 3D reconstructions of the relevant anatomy that are better exploited through full 3D interaction techniques. Although a number of systems allow manipulation of 3D anatomical models they tend to do so through the standard two degrees of freedom available with traditional mouse input. The tracking of hands and gestures in 3D space opens up a much richer set of possibilities for how surgeons manipulate and interact with images through the full six degrees of freedom.

Moreover, with the addition of stereoscopic visualization of 3D renderings, system developers can further address how to enable clinicians to perform new kinds of interactions (such as reaching inside an anatomical model). They might also consider how touchless gestural interaction mechanisms provide new possibilities for interacting with objects and anatomical structures at a distance or otherwise out of reach (such as on a wall-size display and a display unreachable from the operating table). Opportunities do not involve just interaction with traditional theatres and displays but radical new ways to conceive the entire design and layout of operating theatres of the future. C

References

1. Ebert, L., Hatch, G., Ampanozi, G., Thali, M., and Ross, S. Invisible touch: Control of a DICOM viewer with finger gestures using the Kinect depth camera. *Journal of Forensic Radiology and Imaging 1*, 1 (Jan. 2013), 10–14.
2. Ebert, L., Hatch, G., Ampanozi, G., Thali, M., and Ross, S. You can't touch this: Touch-free navigation through radiological images. *Surgical Innovation 19*, 3 (Sept. 2012), 301–307.
3. Gallo, L., Placitelli, A.P., and Ciampi, M. Controller-free exploration of medical image data: Experiencing the Kinect. In *Proceedings of the 24th International Symposium on Computer-Based Medical Systems* (Bristol, England, June 27–30). IEEE Press, 2011, 1–6.
4. Graetzel, C., Fong, T., Grange, S., and Baur, C. A non-contact mouse for surgeon-computer interaction. *Technology and Health Care 12*, 3 (2004), 245–257.
5. Ionescu, A. A mouse in the O.R. *Ambidextrous, Stanford University Journal of Design 4* (June 2006), 30–32.
6. Jacob, M., Li, Y., Akingba, G., and Wachs, J.P. Gestonurse: A robotic surgical nurse for handling surgical instruments in the operating room. *Journal of Robotic Surgery 6*, 1 (Mar. 2012), 53–63.
7. Johnson, R., O'Hara, K., Sellen, A., Cousins, C., and Criminisi, A. Exploring the potential for touchless interaction in image-guided interventional radiology. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, Canada, May 7–12). ACM Press, New York, 2011.
8. Kang, H., Woo Lee, C., and Jung, K. Recognition-based gesture spotting in video games. *Pattern Recognition Letters 25*, 15 (Nov. 2004), 1701–1714.
9. Kipshagen, T., Graw, M., Tronnier, V., Bonsanto, M., and Hofmann, U. Touch- and marker-free interaction with medical software. In *Proceedings of World Congress on Medical Physics and Biomedical Engineering* (Munich, Sept. 7–12). Springer, Berlin, Heidelberg, 2009, 75–78.
10. Leganchuk, A., Zhai, S., and Buxton, W. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer-Human Interaction 5*, 4 (Dec. 1998), 326–359.
11. Mentis, H., O'Hara, K., Sellen, A., and Trivedi, R. Interaction proxemics and image use in neurosurgery. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, TX, May 5–10). ACM Press, New York, 2012, 927–936.
12. Microsoft Corp. *Communicate with computers naturally: Kinect for Windows*; http://www.microsoft.com/en-us/kinectforwindows/
13. Mithun, J., Cange, C., Packer, R., and Wachs, J.P. Intention, context, and gesture recognition for sterile MRI navigation in the operating room. In *Proceedings of CIARP 2012: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Vol. 7441 LNCS* (Buenos Aires, Sept. 3–6, 2012), 220–227.
14. Norman, D. Natural user interfaces are not natural. *ACM interactions 17*, 3 (May–June 2010), 6–10.
15. O'Hara, K., Gonzalez, G., Mentis, H., Sellen, A., Corish, R., and Criminisi, A. Touchless Interaction in *Medical Imaging*. Microsoft Corp., June 2012; http://research.microsoft.com/en-us/projects/touchlessinteractionmedical/
16. Ruppert, G., Amorim, P., Moares, T., and Silva, J. Touchless gesture user interface for 3D visualization using Kinect platform and open-source frameworks. In *Proceedings of the Fifth International Conference on Advanced Research in Virtual and Rapid Prototyping* (Leiria, Portugal, Sept. 28–Oct. 1). Taylor and Francis Group, London, 2011, 215–219.
17. Stern, H., Wachs, J., and Edan. Y. Optimal consensus intuitive hand-gesture vocabulary design. In *Proceedings of the IEEE International Conference on Semantic Computing* (Santa Clara, CA, Aug. 4–7), IEEE Computer Society Press, 2008, 96–103.
18. Strickland, M., Tremaine, J., Brigley, G., and Law, C. Using a depth-sensing infrared camera system to access and manipulate medical imaging from within the sterile operating field. *Canadian Journal of Surgery 56*, 3 (June 2013), E1–6.
19. Tan, J., Chao, C., Zawaideh, M., Roberts, A., and Kinney, T. Informatics in radiology: Developing a touchless user interface for intraoperative image control during interventional radiology procedures. *Radiographics 33*, 2 (Mar.–Apr. 2013), E61–70.
20. Wachs, J., Kolsch, M., Stern, H., and Edan, Y. Vision-based hand-gesture applications. *Commun. ACM 54*, 2 (Feb. 2011), 60–71.
21. Wachs, J., Stern, H., Edan, Y., Gillam, M., Feied, C., Smith, M., and Handler, J. Real-time hand-gesture interface for browsing medical images. *International Journal of Intelligent Computing in Medical Sciences & Image Processing 2*, 1 (June 2008), 15–25.

Kenton O'Hara (keohar@microsoft.com) is a research scientist at Microsoft Research, Cambridge, U.K., and visiting professor in computer science at the University of Bristol, Bristol, U.K.

Gerardo Gonzalez (gerardo.gonzalez_garcia@kcl.ac.uk) is a research fellow in the Biomedical Engineering Department of Kings College London, U.K.

Abigail Sellen (asellen@microsoft.com) is a principal researcher at Microsoft Research, Cambridge, U.K., and honorary professor of computer science at the University of Nottingham, Nottingham, U.K.

Graeme Penney (graeme.penney@kcl.ac.uk) is a senior lecturer in the Biomedical Engineering Department of King's College London, U.K.

Andreas Varnavas (andreas.varnavas@kcl.ac.uk) is a research fellow in the Biomedical Engineering Department of King's College London, U.K.

Helena Mentis (mentis@umbc.edu) is an assistant professor in the Department of Information Systems of the University of Maryland, Baltimore, MD.

Antonio Criminisi (antcrim@microsoft.com) is a senior researcher at Microsoft Research, Cambridge, U.K.

Robert Corish (rcorish@microsoft.com) is a design researcher at Microsoft Research, Cambridge, U.K.

Mark Rouncefield (m.rouncefield@lancaster.ac.uk) is a senior lecturer in the School of Computing and Communications at Lancaster University, Lancaster, U.K.

Neville Dastur (neville@clinsoftsolutions.com) is a locum vascular consultant at Frimley Park Hospital NHS Foundation Trust, Frimley, U.K.

Tom Carrell (tom.carrell@kcl.ac.uk) is a consultant vascular surgeon at Guy's and St. Thomas' NHS Foundation Trust, London, U.K., and honorary senior lecturer in King's College London, U.K.

DOI:10.1145/2500881

The Japanese government tweeted to calm public fear, as the public generally listened to tweets expressing alarm.

BY JESSICA LI, ARUN VISHWANATH, AND H. RAGHAV RAO

Retweeting the Fukushima Nuclear Radiation Disaster

MICRO-BLOGGING LETS THE public stay involved in risk communication following disasters. Here, we explore the patterns of risk communication on Twitter regarding the nuclear radiation threat in the aftermath of the 2011 Fukushima earthquake and tsunami, focusing on patterns of retweets of alarm and reassurance and providing insight into micro-blogging behavior and its consequences.

Communication is important in emergency response.[5,10,11,15,16,18] In the aftermath of the 2011 Fukushima earthquake and tsunami, Twitter was an important social medium for distributing information to millions of people worldwide, including in Japan, with the Japanese government sharing emergency information, relief organizations sharing shelter information, and ordinary citizens posting news of their local situations.[25]

Radiation fears were of utmost importance to the Japanese people. However, the Japanese government and Tokyo Electric Power Company (TEPCO), owner of the crippled nuclear plant in Fukushima, had trouble communicating with them regarding the related risk. Moreover, available information included conflicting and contradictory statements and claims. For example, Greenpeace said data from its own scientists largely correlated with official Japanese data,[8] while Japanese public broadcasting outlet NHK reported "The Japanese government withheld the release of data showing that levels of radiation more than 18 miles (30 kilometers) from the crippled Fukushima nuclear plant exceeded safe levels."[14] The Japanese public's response was to begin asking whether the government was indeed telling the truth or perhaps covering up potential risks that could prompt public panic.

Pew Research reported that Twitter included more than 500 million users worldwide as of February 2012 (http://www.pewinternet.org). Among U.S. Internet users, 15% used the

» key insights

- **Risk communication can be viewed through the lens of the Twitterverse, particularly in the form of retweets, as reflected in the Japanese public's fears over the 2011 Fukushima radiation emergency and the Japanese government's related effort to calm them.**
- **In the days following the disaster, the public was more concerned about the dead and missing and the tsunami's devastation, but fear over the safety of the affected nuclear power plant quickly came to dominate.**
- **While government sources produced more reassuring tweets than did ordinary citizens, they were eventually retweeted less, signifying their loss of influence.**

IMAGE COLLAGE BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES


Japan's chief cabinet secretary says it could be several months before radiation stops leaking from Fukushima nuclear plant.
112,000 Americans die from radiation in Japan.
No one has died from radiation in Japan.
Radiation levels keep rising. Why Japan bury the reactors now?!
Radiation in water rushing into sea tests millions of times over limit.
Japan races to find radiation leak path.
Level of radiation in ocean off Japan's damaged nuclear plant rises to 4,385 times the standard.
Despite everything you've heard, the health risk of radiation in Japan remains pretty low.
Japan eases restrictions on milk, spinach after radiation levels fall below legal limits for three straight weeks.
IAEA says radiation levels at Iitate, 40km from Japan's Fukushima plant, exceed one of its "operational criteria for evacuation."
Three Mile Island was 32 years ago. I was 10 miles away. So here's facts on radiation, beyond the XKCD graphic.
Japan's chief cabinet secretary says it could be several months before radiation stops leaking from Fukushima nuclear plant.
At this time, Japan radioactive particles detected around the world have no health risk to humans.
Governor says radiation levels in Tokyo 20 times normal, The Japan Times reports.
EPA boosts radiation monitoring after low levels found in milk.
The Fukushima situation has exposed a grave danger to our world. But the danger isn't radiation, it's the poor state of science education.
Iodized salt doesn't have enough iodine to protect you from radiation. Too much iodized salt can cause poisoning.
For the first time, Japan sets a standard for the amount of radiation allowed in fish.
Radiation scare sparks run on bottled water in Tokyo.
Radiation counters sell out amid U.S. fears over Japan's nuclear reactor.
Hiroshima organizes scientific teams and medical treatment to aid victims of radiation poisoning
Japan seeks Russia's help over nuclear leak.
CLOSED
TEPCO

Top 25 most retweeted messages.

Green are messages originating with the Japanese government; blue originate from other sources.

Rank	Main Message of Retweet	No. of Retweets	Date of First Tweet	Main Source
1	Low levels of radiation found in U.S. milk.	389	Mar. 31	AP
2	Japan's chief cabinet secretary says it could be several months before radiation stops leaking from Fukushima nuclear plant.	234	Apr. 3	AP
3	Radiation in water rushing into sea tests millions of times over limit.	181	Apr. 5	CNN
4	Despite everything you've heard, the health risk of radiation in Japan remains pretty low.	151	Apr. 7	Time
5	At this time, Japan radioactive particles detected around the world have no health risk to humans.	145	Mar. 25	WHO News
6	Radiation levels keep rising. Why doesn't Japan bury the reactors now?!	130	Mar. 31	Eric Grill
7	Level of radiation in ocean off Japan's damaged nuclear plant rises to 4,385 times the standard.	129	Mar. 31	CNN
8	Three Mile Island was 32 years ago. I was 10 miles away. So here's facts on radiation, beyond the XKCD graphic.	123	Mar. 29	Anil Dash
9	Iodized salt doesn't have enough iodine to protect you from radiation. Too much iodized salt can cause poisoning.	111	Mar. 17	WHO News
10	Radiation counters sell out amid U.S. fears over Japan's nuclear reactor.	108	Mar. 29	WWB
11	The Fukushima situation has exposed a grave danger to our world. But the danger isn't radiation, it's the poor state of science education.	89	Mar. 21	Damned Facts
12	EPA boosts radiation monitoring after low levels found in milk.	87	Mar. 31	CNN
13	Japan eases restrictions on milk, spinach after radiation levels fall below legal limits for three straight weeks.	87	Apr. 8	CNN
14	For the first time, Japan sets a standard for the amount of radiation allowed in fish.	86	Apr. 5	Breaking News
15	Governor says radiation levels in Tokyo 20 times normal, The Japan Times reports.	83	Mar. 16	CNN
16	Hiroshima organizes scientific teams and medical treatment to aid victims of radiation poisoning.	61	Mar. 17	Democracy Now
17	Japan races to find radiation leak path.	57	Apr. 4	BBC
18	Radiation scare sparks run on bottled water in Tokyo.	57	Mar. 24	Reuters
19	IAEA says radiation levels at Iitate, 40km from Japan's Fukushima plant, exceed one of its "operational criteria for evacuation."	56	Mar. 30	BBC
20	Seawater radiation measured at 7.5 million times legal limit.	53	Apr. 5	
21	Fukushima radiation found in U.K.	52	Mar. 29	BBC
22	Japan's chief cabinet secretary Yukio Edano says it could be several months before radiation stops leaking from Fukushima nuclear plant.	51	Apr. 3	Sky News
23	112,000 Americans die from obesity every year. No one has died from radiation in Japan.	51	Mar. 29	Invisible Gaijin
24	Radiation levels "10,000x higher than normal" prompt fears of nuclear reactor breach at Fukushima.	50	Mar. 25	Al Jazeera
25	Japan seeks Russia's help over nuclear leak.	50	Apr. 5	Al Jazeera

service, with 8% doing so on a daily basis. A common Twitter behavior called "retweeting" involves users passing on or sharing the tweets they find of interest to their own followers and contacts. Among the millions of messages disseminated each day, retweets reflect the information that most strongly affects Twitter readers. Retweeting implies at least one reader viewed the information as important enough to want to share it, making retweets more reflective of and influential on the mood of the general population, which hears from a much greater number of observers than it would through a typical lone, unrepeated Twitter message (http://www.retweetrank.com). Our focus here is on how retweets were used to spread both alarming and reassurance information.

Research Background

Use of social media by any government limits the lag in information flow from traditional mass media to individuals and from individuals to other individuals in the form of models (such as the "two-step flow" of communication).[9] Social media offers the government direct access to individuals and other private entities, letting them quickly transfer information in risk communication and information management, disseminating messages of assurance and comfort to the victims of disasters. Risk communication is the focus of information management immediately after a disaster. Certain disaster situations result in prolonged periods of danger and instability as organizations and governments decide how to balance the level of information they should provide, addressing concern over causing panic, and, in some instances, their own reputations and

political need to save face. The nuclear radiation danger in Japan in 2011 was one such instance.[7] What, when, and how to communicate to the public sums up the concept of risk communication.

"Risk communication has been defined as an interactive process of an exchange of information involving multiple messages about the nature of risk," according to Chartier and Gabler.[4] News and social media play an important role not only providing information but also bringing the public's attention to urgent matters. However, when experiencing a stressful situation, the public might view risk-related information issued by official sources (such as a government) as biased, incomplete, or incorrect, tending to leave them feeling dissatisfied.[4]

Reassurance. Many major international news outlets reported the Japanese government withheld release of accurate radiation data,[13] possibly to avoid alarming the public, as it tried to provide reassurance. Prior research explored reassurance, though not in the context of micro-blogging; for example, Barnett et al.[1] weighed the public perception of precautionary advice originating from the U.K. government in 2007 on possible health risks from mobile phones, observing whether risk was presented as less serious than it truly was, leading to public anxiety, panic, or anger when the truth was ultimately revealed. Other times it can be important for a government to attempt to purposely get people concerned. An important matter to understand is how the Twitter environment fosters a balance between alarm and reassurance and how the public receives messages from the government under various circumstances. Ungar[24] explored media reassurance under "hot crisis" conditions, trying to identify the scenarios in which reassuring coverage is more likely than alarming coverage, and found reassurance much more likely in the immediate aftermath of a disaster when panic can spread quickly.

Analysis

We emphasize how retweets echo through the Twitterverse to determine whether the information shared in the aftermath of the Japanese earthquake and tsunami produced reassurance or alarm. Emphasizing the tone of the retweets, we explore what information the public shared. We also explore how the Japanese government communicated through Twitter and how the information was received in the context of reassurance despite the risky situation. Weighing Twitter use in this context will improve how it is managed by governments, as well as by the public, in future disasters.

Data collection. Using the Twitter search API with a service called "Twapper Keeper," we collected several hundred thousand tweets containing the term "Japan radiation" in the month following the earthquake. "RT @" and "via username" were the two most common ways we used to distinguish retweets from regular tweets, helping us narrow the dataset to 38,300 retweets regarding radiation in the month following the earthquake. While it was possible that retrieving tweets based on "Japan radiation" might not have been about 2011, the fact that the data was collected at the time of the incident gave us confidence as to the accuracy of the sample. We did not include other alternative conventions for retweets in our investigation. We also did not clean the data to account for bots, though our inspection of the most frequently cited users in our raw data found none were retweeting and thus not included. We identified and focused on only the top retweeted messages.

Coding schema. To determine the best way to measure characteristics of alarm and reassurance of our included retweets, we examined the previous literature; for example, Stephens and Edison[23] used a simple classification "measuring the number of reassuring or positive statements versus alarming or negative statements," and Speckens et al.[20] developed questionnaires for measuring reassurance (in the context of whether patients feel reassured by their physicians). We adapted some of it to our own research context to develop our coding schema; for instance, reassurance and alarm are not the only dimensions a retweet can have. Likewise, a tweet not expressing alarm is not by default reassuring and vice versa. Following the literature, we created a codebook to help determine if the author of a tweet was communicating alarm, fear, or something else. Coding categories and associated questions included:

Alarm. Did the tweet communicate worry and indicate the situation is dangerous?; and

Reassurance. Did the tweet communicate fear (reverse coded) or communicate calm?

We also measured another dimension:

Doubt. Did the message communicate doubt about the situation?

Finally, we were also interested in whether a particular tweet was from the Japanese government, so we coded two additional questions: Was the

Figure 1. Timeline of radiation retweets.

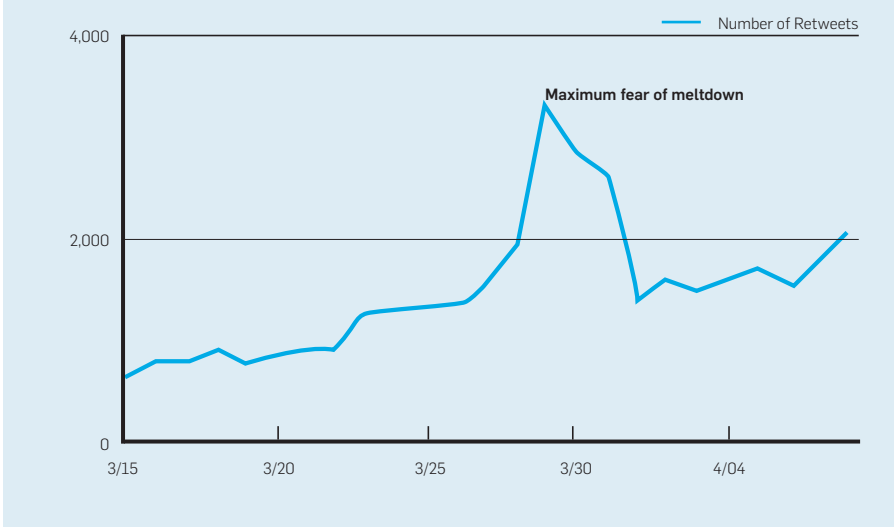
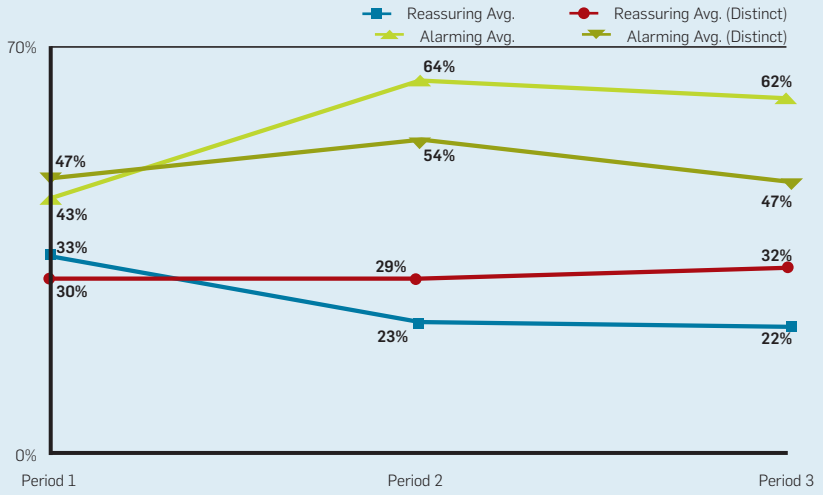


Figure 2. Retweets expressing reassurance and alarm.



message from (or related to) government?, and if it was, did it represent a direct quote or a secondary quote?

Inter-Coder Reliability

We hired two graduate students to work 100 hours each. Both had prior experience on projects involving the coding of tweets. Before the formal coding process began, we conducted a pilot coding session with a randomly chosen 50 tweets. The Kappa value of .90 was high, confirming coding reliability.

Descriptive analysis. Figure 1 outlines the distribution of the retweets for the month following the 2011 earthquake and tsunami. We observed how the retweets and their origin changed over time as the public's concern unfolded. Public interest in the nuclear question began moderately but shifted over time. The public was more focused on the dead and missing and the raw devastation of the tsunami. Meanwhile, the possibility of a nuclear meltdown began to emerge. Indeed, there was still the expectation that cooling of the nuclear plant would resume on March 12 and therefore radiation was a relatively trivial issue in the larger picture of the earthquake and tsunami.[3] However, public fear for the safety of the nuclear plant continued to grow as TEPCO failed to bring the situation under control. The number of retweets with the words "Japan radiation" increased steadily as the situation evolved. Twitter retweet activity peaked in the final days of March as the possibility of a meltdown continued and the public's radiation fear increased dramatically.[26]

Reflecting evolving risk communication, the table here lists the top 25 most commonly retweeted messages and their frequency in the month following the earthquake and tsunami. Messages originating with the government are highlighted in green; others are in blue. Nine conveyed information from a government agency, or 30% of the most frequently retweeted content. The rest was of a more independent nature communicated by a variety of sources. Of the 25 messages, with the exception of two from Twitter users, 23 had all been passed on from major traditional media.

Among the top 25, only three took a reassuring tone, with most reflecting alarm or caution. The top messages sought to avoid overassurance, including "Low levels of radiation found in U.S. milk"; "Japan's chief cabinet secretary says it could be several months before radiation stops leaking from Fukushima nuclear plant"; and "Radiation in water rushing into sea tests millions of times over limit." All managed to only increase public concern. Looking further at the top 25 retweets, at least 12 conveyed alarm and five reassurance. Given the information regarding radiation levels and concern expressed in the top retweeted messages, the general public had messages of both alarm and reassurance, reflecting both sides of the picture.

Results

Of the 38,300 retweeted messages in our sample, our two coders looked at 50 random messages to get a sense of the tone of the messages; we then randomly selected 1,520 more through an Excel `rand()` function. The distribution of the distinct retweets in these messages showed a pattern similar to our full universe of tweets based on a standard data-reduction technique used in large qualitative datasets.[12]

The coders then independently coded the full sample dataset of 1,520 messages. The Kappa value for inter-coder reliability of the full sample dataset was .977; Kappa value higher than .70 is viewed as acceptable, rendering our analysis rigorous since most other such efforts involve inter-coder reliability for a subsample smaller than ours. Note "distinct retweets" refers to this total pool of 1,520 messages, weighing each message equally. "Total retweets" means the number of times each message was retweeted, or the 9,545 times the distinct messages were retweeted in total.

Our coders coded the questions independently; a positive response to one question did not necessarily require a negative response to another. This way we ensured we were measuring truly different things. We conducted a non-parametric T test for the dimensions. The results (t value = 6.78; prob < 0.0001) indicated a statistically significant difference between the mean values for retweets expressing alarm and retweets expressing reassurance.

The number of tweets concerning nuclear radiation began at a relatively low level, shifting over time. Retweet activity peaked at the end of March when fear of a nuclear meltdown was widespread, after which it decreased gradually through early April. We separated the data sample into three periods reflecting these patterns. The first, period 1, included retweets

(579) from before March 29; the second, period 2, included those (493) from March 30 to April 3; and period 3 included retweets (450) from April 4 to April 8. Despite each including a similar number of retweets, each also represents a distinct time period with regard to the disaster and within the micro-blogging community. In the days following the earthquake, little was changing, and the public had not yet focused on the nuclear situation. Twitter activity surged in period 2 with regard to radiation, as a nuclear meltdown became a distinct possibility. Period 3 followed this surge in interest and activity.

Alarm. We conducted nonparametric analysis—Friedman's nonparametric test in SAS—to determine if the three subsample sets reflected different patterns over time communicating messages of alarm; results included Chi-Square = 28.9036 with $p<0.0001$, meaning the sub-datasets showed significant differences in the communication patterns conveying messages of alarm. These results also reflect local Fukushima circumstances as the situation unfolded. The great surge in Twitter activity coincided with a notable increase in alarm. Looking at the average of the two alarm questions in period 1, we see these tweets were unamplified, representing 43% of the total number of messages but 47% of the pool of distinct retweets, an amplification factor of 0.91, or 43%/47%. As alarm increased, the Twitter community began retweeting messages expressing alarm more frequently than those expressing lack of concern, thus amplifying content involving alarm. We calculated an amplification factor of 1.19 in period 2 (64%/54%) and 1.32 in period 3 (62%/47%) (see Figure 2).

As the Fukushima situation grew worse, amplification increased greatly. As alarm information decreased, the public remained in a state of alarm, with its total proportion of retweets elevated at nearly the same level. Although there were fewer messages of alarm for them to retweet, they continued to do so.

Reassurance. We performed nonparametric analysis, with results—Friedman's Chi-square=14.8925, $p=0.0019$—showing "reassuring" communication patterns differed over the three periods.

As the situation continued to worsen, the public retweeted reassuring content less frequently. During period 2, at the end of March, as fear of a meltdown peaked, messages communicating reassurance decreased substantially. A similar level of low reassurance remained through period 3, reflecting that the Japanese public was on edge as the possibility of a meltdown continued despite more reassuring information from the government and media.

The content of the retweets showed a similar level of reassurance. However, the total portion of public retweets dropped in periods 2 and 3, reflecting non-amplification of the information. The most frequently retweeted messages expressed alarm, while messages expressing reassurance were retweeted less frequently. The public had stopped echoing the reassuring information—the exact opposite of what was seen in the alarm dimension when the community was quick to echo alarm content.

All three periods showed significantly more alarm retweets in terms of number of distinct retweets and number of retweets. Moreover, there appeared to be an inverse relationship between proportion of retweets of alarm and of reassurance.

Figure 3. Distinct vs. total retweets originating with the Japanese government.

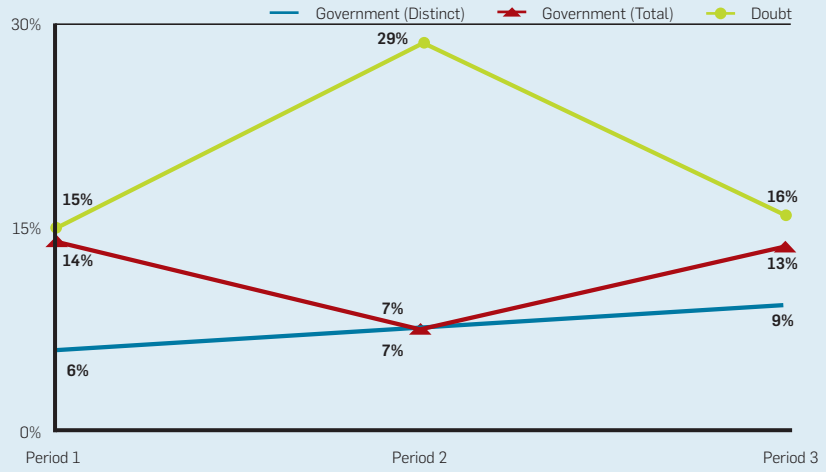
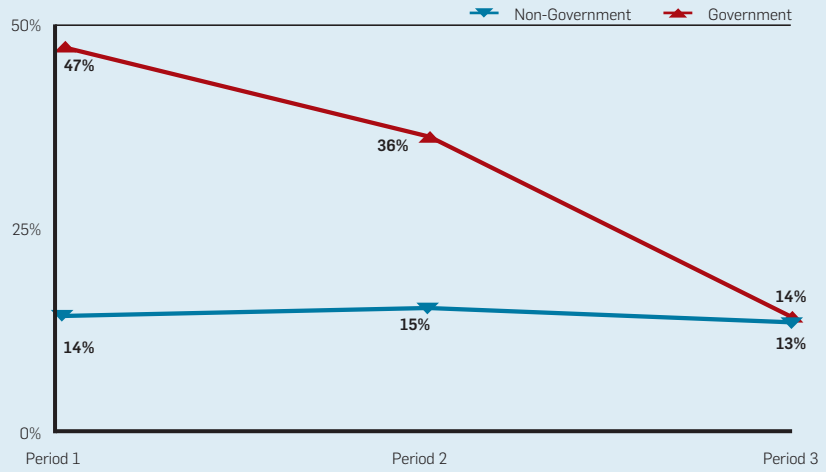


Figure 4. Messages communicating reassurance.



Messages from the government. In period 1, prior to the surge in Twitter activity, the government's messages enjoyed significant amplification from the Twitter community. Despite accounting for only 5.7% of the messages being retweeted, the number of times the messages were retweeted accounted for over 14% of the total retweets in our sample of 1,520 retweeted messages, resulting in an amplification factor of 2.5 (14%/5.7%). However, this amplification vanished completely in period 2. The percentage of distinct retweets and percentage of the total retweets from the government were identical at 7.4%, an amplification factor of 1 (see Figure 3).

The voice of the government was either drowned out or ignored by the Twitter community, so, at least in terms of Twitter, had become less influential. Messages originating with the government represented a slightly growing percent of the sample of distinct messages retweeted over time, though they were retweeted proportionally less frequently as interest intensified. The government's aim was to get more information to the public, but this did not translate into information being distributed more widely. Moreover, the public did not echo the additional information from the government. Amplification of government information partially returned in period 3 when the government was the source of 9.2% of distinct retweets and 13.4% of total retweets. This 46% amplification was far short of the 145% amplification of government tweets in period 1.

Another question we asked was whether a particular retweet communicated doubt. We observed a significant increase in doubt in period 2 that then decreased in period 3. The decline in amplification of governmental information directly corresponded to this increase in doubt. We theorized that as public doubt increased, the public was less likely to amplify the government's information by repeating it. We drew a similar conclusion regarding how direct quotation of government sources dropped significantly as doubt increased.

Government tweets. While the government had no control over the frequency of its words that were retweeted, it was able to control the amount and content of information it provided. In period 1, 47% of government messages communicated calm, while only 14% of non-government retweeted messages communicated calm (see Figure 4). In period 2, 36% of government messages communicated calm, though it was only 15% of the non-government sample.

As public doubt increased, the public was less likely to amplify the government's information by repeating it.

Discussion

This analysis reflects strong evidence that the alarming and reassuring tone of the micro-blogging environment changes dramatically as a situation and the public's mind-set evolve. We also found the information communicated through retweeted Twitter messages can be notably different from that identified in previous research of television and newspaper coverage;[23] while that research found traditional media coverage was more reassuring, the micro-blogging universe generally expressed more alarm.

Nevertheless, we noticed a large number of the most retweeted messages originated from traditional media sources, showing how traditional media have embraced this new channel. The effect of the traditional media's voice is reflected in the high frequency of their content in retweeted data.

We found retweets from governmental sources reflected a much more reassuring message than those not coming from the government, especially in the immediate aftermath of the earthquake (and nuclear disaster) when fear of a meltdown was pervasive among the general public. In order to calm potential panic, government agencies and sources may sometimes look to withhold or postpone releasing negative data, especially when their own reputations and face saving are involved. Over time, government feels less of a need to reassure, shifting toward a more alarmed posture. While governments initially wish to avert panic, they ultimately need to keep the public alert and focused on possible danger.

Conclusion

Our results point to the need for reassuring messages via social media, as well as their value in risk communica-

tion, as the micro-blogging universe takes a more alarmed tone compared to traditional media. Subsequent disasters (such as Hurricane Sandy in 2012 on the East Coast of the U.S.) show such reassurance can come not only from the government but also from the private sector; for example, in the aftermath of Sandy in New York City and along the New Jersey coast, Con Edison, the regional electric utility, sent reassuring Twitter messages regarding return of power on particular streets and buildings and estimates of when power would be restored to the rest.[22] It did a good job engaging the general public, reassuring it and showing how social media can play a role as important as or more important than word of mouth. However, tweets and retweets are extremely valuable when the major communication channels are down or difficult to access. Public-private partnerships are needed to rebuild communities ruptured during such situations, with social media playing an especially important role.

In this article, we have focused on retweets rather than whether news was broken first on Twitter. The former has received limited attention from the perspective of disaster preparedness and remains a subject for future exploration. Moreover, Twitter is unlike other social media in that it restricts a message to 140 characters; how this inhibits or fosters the exchange of information during emergencies remains unclear and is thus likewise a subject for future exploration. Another issue not addressed here is motivation; for example, why did individuals retweet certain news stories more than others? The secondary nature of our analysis restricted examination of this question because it requires polling retweeters for such motivations and is also likewise a subject for future exploration.

Our current research focused solely on retweets on Twitter rather than on other social media platforms. Perhaps other platforms (such as Facebook) also experienced similar (or different) communication behaviors. Without data either way, our study was limited to Twitter users and followers. Moreover, because it did not focus on the effectiveness of retweets, we did not code the popularity of respective Twitter messages. Hence, a retweet from a Twitter user with many followers might perhaps be more effective than one from one with fewer followers. Finally, we did not know or code for the geographical information of the retweeters. It would be interesting to see how retweets from different geographical locations differ with respect to the geographical epicenter of a disaster, representing limitations of our study and topics for potential future research.

Acknowledgment

This research was funded in part by the National Science Foundation under grants 0916612 and 1134853. The research of the third (corresponding) author, H. Raghav Rao, was supported in part by Sogang Business School's World Class University Project (R31-20002) funded by the Korea Research Foundation and by the Sogang University Research Fund. Any opinions, findings, and conclusions or recommendations expressed here are those of the researchers and do not necessarily reflect the views of the National Science Foundation; NSF has not approved or endorsed this content.

References

1. Barnett, J., Timotijevic, L., Shepherd, R., and Senior, V. Public responses to precautionary information form the Department of Health (U.K.) about possible health risks from mobile phones. *Health Policy 82*, 2 (Nov. 2007), 240–250.
2. Baroudi, J.J., Olson, M.H., Ives, B., and Davis, G. An empirical study of the impact of user involvement on system usage and information satisfaction. *Commun. ACM 29*, 3 (Mar. 1986), 232–238.
3. Bendeich, M. Timeline: Japan's unfolding nuclear crisis. *Reuters* (Mar. 16, 2011); http://www.reuters.com/article/2011/03/16/japan-quake-timeline-idUSL3E7EG3GA20110316
4. Chartier, J. and Gabler, S. *Risk Communication and Government: Theory and Application for the Canadian Food Inspection Agency.* Public and Regulatory Affairs, Canadian Food Inspection Agency, 2000; http://www.aphis.usda.gov/oieamericas/riskcomm.pdf
5. Chen, R., Sharman, R., Rao, H.R., and Upadhyaya, S.J. Coordination in emergency response management. *Commun. ACM 51*, 5 (May 2008), 66–73.
6. Iacovou, C.L. and Nakatsu, R. A risk profile of offshore-outsourced development projects. *Commun. ACM 51*, 6 (June 2008), 89–94.
7. Iizuka, S., Ogawa, K. et al. Factors affecting user reassurance when handling information in a public work environment. *International Journal of Human-Computer Interaction 23*, 1/2 (Dec. 2007), 163–183.
8. Jolly, D. Japanese operator says it will scrap four reactors at plant. *New York Times* (Mar. 30, 2011); http://nuclearno.com/text.asp?15178
9. Katz, E. and Lazarsfeld, P.F. *Personal Influence.* Free Press, New York, 1955.
10. Kim, M., Sharman, R., Cook-Cottone, C.P., Rao, H.R., and Upadhyaya, S.J. Assessing roles of people, technology, and structure in emergency management systems: A public-sector perspective. *Behaviour & Information Technology 31*, 12 (Dec 2012).
11. Li, J. and Rao, H.R. Twitter as a rapid response news service: An exploration in the context of the 2008 China earthquake. *The Electronic Journal of Information Systems in Developing Countries 42*, 4 (July 2010), 1–22.
12. Namey, E., Guest, G., Thairu, L., and Johnson, L. *Handbook for Team-Based Qualitative Research*, G. Guest and K.M. MacQueen, Eds. Alta Mira Press, New York, 2008
13. Newscore staff report. Japan accused of cover-up, plans to dump radioactive water in Pacific. Apr. 4, 2011; http://www.dailytelegraph.com.au/japan-accused-of-nuclear-cover-up/story-fn6e1m7z-1226033714858
14. Post staff report. Japanese government withheld information about high radiation levels. *New York Post* (Apr. 4, 2011); http://nypost.com/2011/04/04/japanese-government-withheld-information-about-high-radiation-levels/
15. Oh, O., Agrawal, M., and Rao, H.R. Information control and terrorism: Tracking the Mumbai terrorist attack through Twitter. *Information Systems Frontiers (Special Issue on Terrorism Informatics) 13*, 1 (Mar. 2011), 33–43.
16. Oh, O., Kwon, K.H., and Rao, H.R. An exploration of social media in extreme events: Rumor theory and Twitter during the Haiti earthquake 2010. In *Proceedings of the International Conference on Information Systems* (St. Louis, MO, Dec. 12–15, 2010).
17. Palen, L., Hiltz, S.R., and Liu, S. Online forums supporting grassroots participation in emergency preparedness and response. *Commun. ACM 50*, 3 (Mar. 2007), 54–58.
18. Savage, N. Twitter as medium and message. *Commun. ACM 54*, 3 (Mar. 2011), 18–20.
19. Slovic, P. Trust, emotion, sex, politics, and science: Surveying the risk-assessment battlefield. *Risk Analysis 19*, 4 (1999), 689–701.
20. Speckens, A.E.M., Spinhoven, P. et al. The reassurance questionnaire: Psychometric properties of a self-report questionnaire to assess reassurability. *Psychological Medicine 30*, 4 (July 2000), 841–847.
21. Starbird, K. and Palen, L. Pass it on?: Retweeting in mass emergency. In *Proceedings of the Conference on Information Systems for Crisis Response and Management* (Harbin, China, Aug. 2010).
22. Spall, M.J. Hurricane Sandy: The million-customer storm. At Symposium on Super Storm Sandy: Lessons Learned (John Jay College, New York, Feb. 21, 2013).
23. Stephens, M. and Edison, N.G. News media coverage of issues during the accident at Three-Mile Island. *Journalism Quarterly 59*, 2 (June 1982), 199–259.
24. Ungar, S. Hot crises and media reassurance: A comparison of emerging diseases and Ebola Zaire. *The British Journal of Sociology 49*, 1 (Mar. 1998), 36–56.
25. Wallop, H. Japan earthquake: How Twitter and Facebook helped. *The Telegraph* (Mar. 13, 2011); http://www.telegraph.co.uk/technology/twitter/8379101/Japan-earthquake-how-Twitter-and-Facebook-helped.html
26. Yamazaki, T. and Maruta, F. Tokyo Electric reviewing water tests showing iodine at 10,000 times limit. *Bloomberg* (Mar. 31, 2011); http://www.bloomberg.com/news/2011-03-31/japan-reviewing-water-tests-showing-iodine-at-10-000-times-limit.html

Jessica Li (puli@buffalo.edu) is a vice president of Citigroup, Institutional Client Group, in New York; her work on this article was done while she was in the School of Management of the State University of New York at Buffalo.

Arun Vishwanath (avishy@buffalo.edu) is an associate professor in the Department of Communication of the State University of New York at Buffalo.

H. Raghav Rao (mgmtrao@buffalo.edu) is a SUNY distinguished service professor in the Management Science and Systems Department of the School of Management and an adjunct professor in the Computer Science and Engineering Department of the State University of New York at Buffalo.

DOI:10.1145/2541883.2541900

Control transactions without compromising their simplicity for the sake of expressiveness, application concurrency, or performance.

BY VINCENT GRAMOLI AND RACHID GUERRAOUI

Democratizing Transactional Programming

THE TRANSACTION ABSTRACTION encapsulates the mechanisms used to synchronize accesses to data shared by concurrent processes, dating to the 1970s when proposed in the context of databases to ensure consistency of shared data.[7] This consistency was determined with respect to a sequential behavior through the concept of serializability;[25] concurrent accesses must behave as if executing sequentially or be atomic. More recently, researchers have derived other variants (such as opacity[13] and isolation[30]) applicable to different transactional contexts.

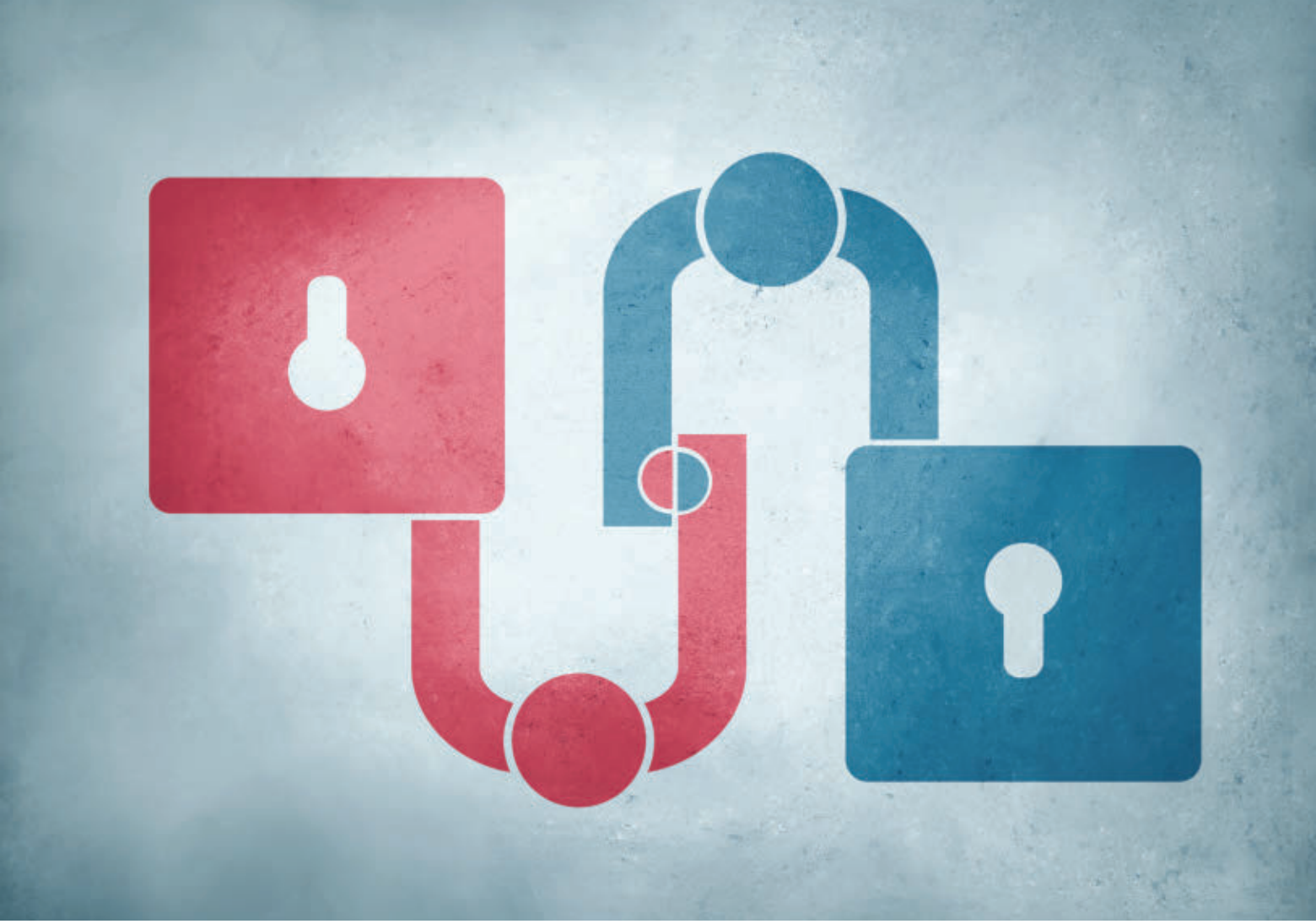
The transaction abstraction was first considered as a programming language construct in the form of guards and actions by Liskov and Scheifler more than 30 years ago,[22] then adapted to various programming models, including Eden,[1] ACS,[12] and Argus.[21] The first hardware support for a transactional construct was proposed in 1986 by Tom Knight,[19] basically introducing parallelism in functional languages by providing synchronization for multiple memory words. Later, the notion of transactional memory was proposed in the form of hardware support for concurrent programming to remedy the trickiness and subtleties of using locks (such as priority inversion, lock-convoying, and deadlocks)[18] (see Figure 1).

Since the advent of multicore architectures approximately 10 years ago, the very notion of transactional memory has become an active topic of research (http://www.cs.wisc.edu/trans-memory/biblio/list.html). Hardware implementations of transactional systems[18] turned out to be limited by specific constraints programmers could "abstract away" only through unbounded hardware transactions. However, purely hardware implementations are complex solutions most industrial developers no longer explore. Rather, a hybrid approach was adopted through a best-effort hardware component that must be complemented by software transactions.[4]

Software transactions were originally designed in the mid-1990s as a reusable and composable solution to execute a set of shared memory accesses fixed prior to execution.[29] More recently, they were applied to handle when the control flow is not predetermined.[17] Early investigations of the performance of software transactions questioned their ability to leverage multicore architectures.[2] However, these results were revisited by Dragojevic et al.,[6] showing a highly optimized software-transactional

» key insights

- **Though powerful, the transaction paradigm can sometimes restrict application concurrency and performance.**
- **Democratizing transactional programming aims to make the paradigm useful for novice programmers who want concurrency to be transparent and for expert programmers who are able to address its underlying challenges while compromising neither composition nor correctness.**
- **The key challenge is how to offer multiple synchronization semantics of sequences of shared data accesses without compromising safety and liveness.**



memory (STM) with manually instrumented benchmarks and explicit privatization whose throughput still outperforms sequential code by up to 29 times on SPARC processors with 64 concurrent threads and by up to nine times on x86 with 16 concurrent threads. However, performance remains the main obstacle preventing wide adoption of the transaction abstraction for general-purpose concurrent programming.

In classic form, transactions prevent expert programmers from extracting the same level of concurrency possible through more primitive synchronization techniques. This observation is folklore knowledge, yet we show for the first time, in this article through a simple example, that this limitation is inherent in the transaction concept in its classic form irrespective of how it is used. It can be viewed as the price of bringing concurrency to the masses and making it possible for average programmers to write parallel programs that use shared data. Nevertheless, some programmers are indeed concurrency experts and might find it frustrating if they are not able to use their skills to enhance concurrency and performance.

Not surprisingly, researchers have been exploring relaxation of the classic transaction model[23,24,27] that enables more concurrency. Doing so while keeping the simplicity of the original model has proved to be a challenge; the idea is to preserve the original sequential code while composing applications devised by different programmers, possibly with different skills.

Here, we endorse mixing different transaction semantics within the same application, with strong semantics to be used by novice programmers and weaker semantics by concurrency experts. The challenge is to ensure the polymorphic system mixing different semantics still enables code reuse, composing it in a smooth manner. Before describing how such mixing can be addressed, we take a closer look at the meaning of reuse and composition.

Inherent Appeal of Transactions

The transaction paradigm is appealing for its simplicity, as it preserves sequential code and promotes concurrent code composition.

Algorithm 1. An implementation of a linked list operation with transactions

```
1: tx-contains(val)_p:
2:   int results;
3:   node *prev, *next;
4:   transaction {
5:     curr = set → head;
6:     next = curr → next;
7:     while next → val < val do
8:       curr = next;
9:       next = curr → next;
10:    result = (next → val ==
val);
11: }
12: return result;
```

Preserving sequentiality. Transactions preserve the sequential code in that their use does not alter it beyond segmenting it into several transactions. More precisely, the regions of sequential code that must remain

ILLUSTRATION BY BY ALICIA KUBISTA/ANDRIJ BORYS ASSOCIATES

atomic in a concurrent context are simply delimited, typically by a `transaction{...}` block, as depicted in Algorithm 1; the original structure depicted in Algorithm 2 remains unchanged.

Programming with transactions shifts the inherent complexity of concurrent programming to implementation of the transaction semantics that must be done once and for all. Due to transactions, writing a concurrent application follows a divide-and-conquer strategy where experts write a live, safe transactional system with an unsophisticated interface, and the novice writes a transaction-based application or delimits regions of sequential code.

Algorithm 2. The linked list node

```
1: Transactional structure node:
2:    intptr_t val;
3:    struct node * next;
4:    //Metadata management is
      implicit
5: Lock-based structure node_lk:
6:    intptr_t val;
7:    struct node_lk * next;
8:    volatile pthread_spinlock_
      tlock;
```

Traditional synchronization techniques generally require programmers first re-factorize the sequential code. Using lock-free techniques, they typically use subtle mechanisms (such as logical deletion[14]) to prevent inconsistent memory de-allocations. Using lock-based techniques, they usually explicitly declare and initialize all locks before using them to protect memory accesses, as in Algorithm 2 line 8.

The transaction abstraction hides both synchronization internals and metadata management. If locks or timestamps are used internally, they are declared and initialized transparently by the transactional system. All memory accesses within a transaction block are transparently instrumented by the transactional system as if they were wrapped. The wrappers can then exploit the metadata, locks, and timestamps to detect conflicting accesses and potentially abort a transaction.

Enabling composition. Transactions allow Bob to compose existing transactional operations developed by Alice into a composite operation that preserves the safety and liveness of its components[15] (see Figure 2).

Alternative synchronization techniques do not facilitate composition. Consider a simple directory abstraction mapping a name to a file. With transactions, a programmer is able to compose the removal of a name and creation of a new name into a `rename` action. If a user `renames` a file from one directory d_1 to another directory d_2 and another user renames a file from d_2 to d_1, directories must be protected to avoid deadlocks; that is, Bob must first understand the locking strategy of Alice to ensure the liveness of his own operations. For this reason, the header of the Linux kernel file `mm/filemap.c` includes 50 lines of comments explaining the locking strategy. Lock-free techniques are even more complex, requiring a multi-word compare-and-swap operation to make the two renaming actions atomic while retaining concurrency.[11]

In contrast, a transactional system detects a conflict between the two renaming transactions and lets only one of them resume and possibly commit; the other is restarted or resumed later. Deciding on a conflict-resolution strategy is the task of a dedicated service, or "contention manager," for which various strategies and implementations have been proposed.[28]

Inherent Limitation of Transactions

A transaction delimits a region of accesses to shared locations and protects the set of locations accessed in this region. By contrast, a (fine-grain) lock generally protects a single location, even though it is held during a series of accesses, as depicted in Algorithm 3. This difference is crucial, as it translates into the differences between transactions and locks in terms of expressiveness, concurrency, and performance.

Lacking expressiveness. To reinforce our point that transactions are inherently limited in terms of expressiveness we define "atomicity" as a bi-

Figure 1. History of transactions.

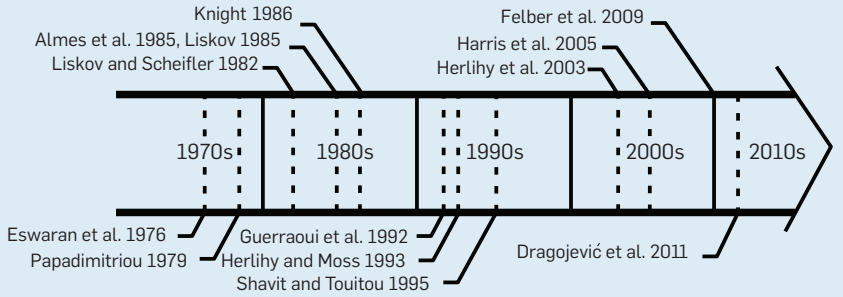


Figure 2. Bob composes Alice's component operations `remove` and `create` into a new operation `rename` that preserves the safety and liveness of its components.

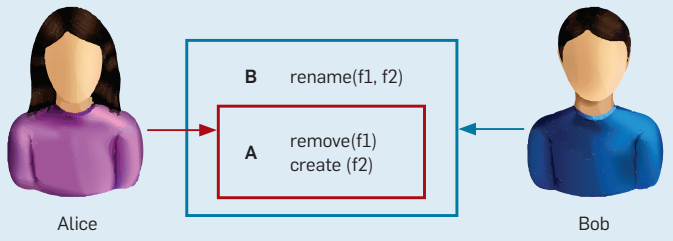
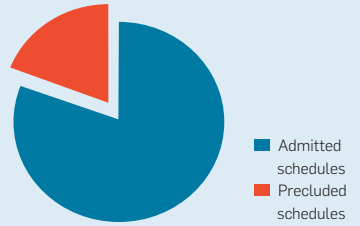


Figure 3. Transactions preclude 20% of the correct schedules of a simple concurrent linked list program.



nary relation over shared memory accesses π and π' of a single transaction within an execution α: *atomicity*(π, π') is true if π and π' appear in α as if both occur at one common indivisible point of the execution. It is important to note this relation is not transitive; that is, *atomicity*(π_1, π_2) $\wedge$ *atomicity*(π_2, π_3) $\not\Rightarrow$ *atomicity*(π_1, π_3).

Algorithm 3. An implementation of a linked list operation with locks

```
1: lk-contains(val)_p:
2:    int results;
3:    node_lk *prev, *next;
4:    lock(&set → head → lock);
5:    curr = set → head;
6:    lock(&curr → next → lock);
7:    next = curr → next;
8:    while next → val < val do
9:       unlock(&curr → lock);
10:      curr = next;
11:      lock(&next → next →
         lock);
12:      next = curr → next;
13: unlock(&curr → lock);
14: result = (next → val ==
    val);
15: unlock(&next → lock);
16: return result;
```

As π_2 may appear to have executed at several consecutive points of the execution, the points at which π_1 and π_2 appear to have occurred may be disjoint from the points at which π_2 and π_3 appear to have occurred.

A process locking x (with mutual exclusion) during the point interval $(p_1; p_2)$ of α, in which it accesses x guarantees any of its other accesses during this interval will appear atomic with its access to x; for example, in the following lock-based program, where $r(x)$ and $w(x)$ denote (respectively) read and write accesses to shared variable x, process (or more precisely thread) P_ℓ guarantees *atomicity*($r(x)$; $r(y)$) and *atomicity*($r(y)$, $r(z)$) but not *atomicity*($r(x)$, $r(z)$):

$P_\ell = \texttt{lock}(x)\, r(x)\, \texttt{lock}(y)\, r(y)\, \texttt{unlock}(x)\, \texttt{lock}(z)\, r(z)\, \texttt{unlock}(y)\, \texttt{unlock}(z)$.

Conversely, a process P_t executing the following transaction block ensures *atomicity*($r(x)$; $r(y)$), *atomicity*($r(y)$, $r(z)$) but also *atomicity*($r(x)$, $r(z)$), the transitive closure of the atomicity relations guaranteed by P_ℓ. Using classic transactions, there is no way to write a program with semantics similar to P_ℓ or ensure the two former atomicity relations without also ensuring the latter.

$P_t = \texttt{transaction}\{\, r(x)\, r(y)\, r(z)\, \}$.

This lack of expressiveness is not related to the way transactions are used but to the transaction abstraction itself. The open/close block somehow blindly guarantees that all the accesses it encapsulates appear as if there was an indivisible point in the execution where all take effect.

Effect on concurrency. Not surprisingly, the limited expressiveness of transactions translates into a concurrency loss; for example, consider the transactional linked list program in Algorithm 1. Clearly, the value of the *head* → *next* pointer observed by the transaction (line 6) is no longer important when the transaction is checking whether the value *val* corresponds to a value of a node further in the list (line 7), yet a concurrent modification of *head* → *next* can invalidate the transaction when reading *next* → *val*, as transactions enforce atomicity of all pairs of accesses; this is a false-conflict leading to unnecessary aborts. Conversely, the hand-over-hand locking program of Algorithm 3 allows such a concurrent update (line 7) when checking the value (line 8), starting from the second iteration of the while-loop.

To quantify the effect of the limited expressiveness of transactions on the number of accepted schedules, consider a concurrent program where the process P_t executes concurrently with processes $P_1 = \texttt{transaction}\{w(x)\}$ and $P_2 = \texttt{transaction}\{w(z)\}$. As there are four ways to place the single access of one of these two processes between accesses of P_t and five ways to place the remaining one in the resulting schedule, there are 20 possible schedules. Note that all are correct schedules of a sorted linked list implementation.

However, most transactional memory systems guarantee each of their executions is equivalent to an execution where sequences of reads and writes representing transactions are executed one after another (serializability) in an order where no transaction terminating before another start is ordered after (strict-

Performance remains the main obstacle preventing wide adoption of the transaction abstraction for general-purpose concurrent programming.

ness). (This guarantee is often satisfied, as a large variety of transactional memory systems ensures opacity,[13] a consistency criterion even stronger than this strict serializability, as it additionally requires noncommitted transactions never observe an inconsistent state.) These transactional memory systems preclude four of these schedules (see Figure 3): those in which P_t accesses x before P_1 (P_t is serialized before P_1, or $P_t \prec P_1$), P_1 terminates before P_2 starts ($P_1 \prec P_2$) and in which P_2 accesses z before P_t ($P_2 \prec P_t$). This limitation translates here into concurrency loss.

Worth noting is that a programmer could exploit weaker transactional memory systems to export these serializable histories.[10,26] Such systems would offer a transaction that might not be appropriate for all possible uses; for example, it might be possible that one transaction reads an inconsistent state before aborting. In fact, the concurrency limitation is due to transactional memory systems providing a unique but general-purpose transaction.

Effect on performance. The metadata management overhead of software transactions when starting, accessing shared memory, and committing is typically expected by the programmer to be compensated by exploiting concurrency.[6] In scenarios like the linked list program outlined earlier where transactions fail to fully exploit all available concurrency, their performance cannot compete with other synchronization methodologies. Recall this is due to the expressiveness limitation inherent in transactions; the limitation is thus not tied to the way transactions are used but to the abstraction itself.

To depict the effect on performance, we compared the existing Java concurrency package to the classic transaction library TL2[5] on a 64-way Niagara 2 SPARC-based machine. Note this is the Java implementation of the TL2 algorithm that detects conflicts at the level of granularity of fields and is distributed within DeuceSTM,[20] a bytecode instrumentation framework offering a suite of TM libraries. We present the results obtained on a simple `Collection` benchmark of 2^{12} elements providing `contains`, `add`, `remove`, and `size` operations with an update ratio and a `size` ratio of 10%, respectively. As the existing lock-free data structures do not support atomic size we had to use the `copyOnWriteArraySet` workaround of this package, comparing it against the linked list implementation building on TL2.

To adequately exploit the concurrency allowed by the semantics of an application, programmers must be willing to trade simplicity for additional control.

Figure 4 uses the throughput (committing transactions per time unit) of the bare sequential implementation (without synchronization) as the baseline, illustrating the throughput speedup (over sequential) a programmer can achieve through either the classic transactions or the existing `java.util.concurrent` package. When its normalized throughput is 1, the throughput of the corresponding concurrent implementation equals the throughput of the sequential implementation. In particular, the graph indicates the existing collection performs 2.2x faster than classic transactions on 64 threads. The poor performance of classic transactions is due to their lack of concurrency, a problem addressed in the next section.

Democratizing Transactions

Traditionally, transactional systems ensure the same semantics for all their transactions, independent of their role in concurrent applications. However, as discussed, these semantics are overly conservative and, by limiting concurrency, could also limit performance. Without additional control, skilled programmers would be frustrated by not being able to obtain highly efficient concurrent programs. To adequately exploit the concurrency allowed by the semantics of an application, programmers must be willing to trade simplicity for additional control.

To be a widely used programming paradigm, the transactional abstraction must be democratized, or universally useful and available to all programmers. Not only should transactions be an off-the-shelf solution for novices, they should also permit additional control to experts in concurrent programming. Simple default semantics should be able to run concurrently with transactions of more complex semantics, capturing more subtle behaviors. The concurrency challenge is twofold: The

transaction abstraction must allow expert programmers to easily express hints about the targeted application semantics without modifying the sequential code, and the semantics of each transaction must be preserved, even though multiple transactions of different semantics can access common data concurrently. This second property, semantics, is crucial but makes development of a transactional system even more complex.

Relaxation and sequentiality. Several transaction models have been proposed as a relaxed alternative to the classic one. Examples are open nesting[24] and transactional boosting.[16] Both exploit commutativity by considering transactional operations at a high level of abstraction. Both also acquire abstract locks to apply nested operations and require the programmer to specify compensating actions or inverse operations to roll back these high-level changes. To avoid deadlocks due to acquisition of new locks at abort time, the programmer may follow lock-order rules or exploit timeouts. Alternatively, other approaches extend the interface of the transactional memory system with explicit mechanisms like functions `light-reads`, `unit-loads`, `snap`, and early `release`; for example, programmers can use early `release` explicitly to indicate from which point of a transaction all conflicts involving its read of a given location can be ignored.[17] The challenge is thus to achieve the same concurrency achievable through these models while preserving sequential code and composition of transactions.

The elastic transaction model[8] aims to preserve sequential code and guarantee composition, providing, together with the classic form of transaction model, a semantics of transactions that enables programmers to efficiently implement search structures. As in a classic transaction, the programmer must delimit the blocks of code that represent elastic transactions, preserving sequential code as depicted in Algorithm 4. Elastic transactions bypass deadlocks by updating memory only at commit time, avoiding the need to acquire additional locks upon abort.

Unlike classic transactions, during execution, an elastic transaction can be cut (by the elastic transactional system) into multiple classic transactions, depending on the conflicts it detects.

Algorithm 4. Java pseudocode of the `add()` operation with elastic transactions

```
1: public boolean add (E e):
2:  transaction(elastic) {
3:   Node(E)prev = null
4:   Node(E)prev = head
5:   E v
6:
7:   if next == null then // empty
8:      head = newNode(E)(e,next)
9:      return false
10:  while (v = next.getVal-
     ue()).compareTo(e) < 0 do
     // non-empty
11:     prev = next
12:     next = next.getNext()
13:     if next == null then
        break
14:  if v.compareTo(e) == 0 then
15:     return false
16:  if prev == null then
17:     Node(E)n = new Node(E)
        (e,next)
18:     head = n
19:  else prev.setNext(new
     Node(E)(e,next))
20:  return true
21: }
```

Consider the following history of shared accesses in which transaction j `adds` 1 while transaction i is parsing the data structure to `add` 3 at its end:

$$\mathcal{H} = r(h)^i, r(n)^i, r(h)^j, r(n)^j, w(h)^j, r(t)^i, w(n)^i.$$

This history is neither serializable[25] nor opaque[13] since there is no history in which transactions i and j execute sequentially and where $r(h)^i$ occurs before $w(h)^j$ and $r(n)^j$ occurs before $w(n^i$; the high-level insert operations of this history are atomic. A traditional transactional scheme would detect two conflicts between transactions i and j and prevent them both to commit. Nevertheless, history $\mathcal{H}$ does not violate the correctness of the integer set; 1 appears to be added before 3 in the linked list, and both are present at the end of the execution.

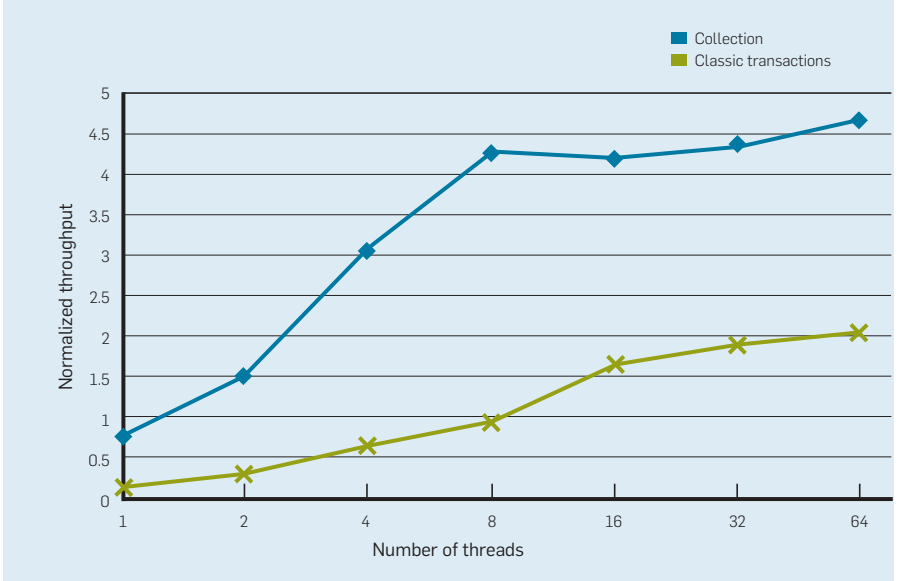
The programmer must label transaction i as being elastic to solve this issue. History $\mathcal{H}$ can then be viewed as the combination of several transactions:

$$f(\mathcal{H}) = [r(h)^i, r(n)^i]^{s1}, r(h)^j, r(n)^j, w(h)^j, [r(t)^i, w(n)^i]^{s2}.$$

In $f(\mathcal{H})$, elastic transaction i is cut into two transactions: s_1 and s_2. Crucial to the correctness of this cut, no two modifications on n and t have occurred between $r(n)^{s1}$ and $r(t)^{s2}$. Otherwise, the transaction would have to abort.

These cuts enable more concurrency than what an expert programmer could accomplish with classic transactions for two main reasons: First, the cuts are tried dynamically at runtime depending on the interleaving of

Figure 4. Throughput (normalized over the sequential throughput) of classic transactions and existing concurrent collection.



accesses; as this interleaving is generally nondeterministic, the programmer cannot just split transactions prior to execution and ensure correct executions. Second, as elastic transactions rely on dynamic information, they exploit more information than static commutativity of operations; for example, elastic transactions enable additional concurrency between two linked list `adds` by allowing the history involving transactions t_1 and t_2: $r(h)^{t1}$, $r(n)^{t2}$, $w(h)^{t2}$, $w(n)^{t1}$ in which neither $r(n)^{t2}$ and $w(n)^{t1}$ nor $r(h)^{t1}$ and $w(h)^{t2}$ commute.

Composition and mixture of semantics. The more semantics the transactional system provides, the more control it gives expert programmers, allowing them to boost performance. The opacity semantics of classic transactions benefit the novice programmer, as they are always safe to use. The elastic transactions can bring added performance in search structures. A programmer can also consider the mix of the opaque classic and the relaxed elastic models with a new semantics we call "snapshot" semantics. This mix is particularly appealing for obtaining (efficiently) a result that depends on numerous elements of a data type (such as a Java Iterator); see, as an example, the snapshot transaction implementing a size method in Algorithm 5.

At first glance, providing as many forms as possible in a single toolbox system may seem to be the key solution for developing concurrent applications, but the challenge involves the mixture of these semantics. Mixing them requires letting them access the same shared data concurrently. It is crucial that the semantics of each individual transaction is not violated by the execution of concurrent transactions of potentially different semantics; for example, the key idea for highly concurrent snapshot semantics is to exploit multi-version concurrency control to let snapshots commit while concurrent (elastic or classic) updates commit. A typical implementation of a snapshot is to exploit a global counter and a version number per written value so the transaction can fetch the counter at start time and decide (while reading new locations) to return a value that has an appropriate (not too recent) version consistent with this start time.

Algorithm 5. Java pseudocode of the `size()` operation with a snapshot transaction

```
1: public int size():
2:  transaction(snapshot) {
3:    int n = 0
4:    Node(E) curr = head
5:
6:    while curr ≠ null do
7:     curr = curr.getNext()
8:     n++
9:    return n
10: }
```

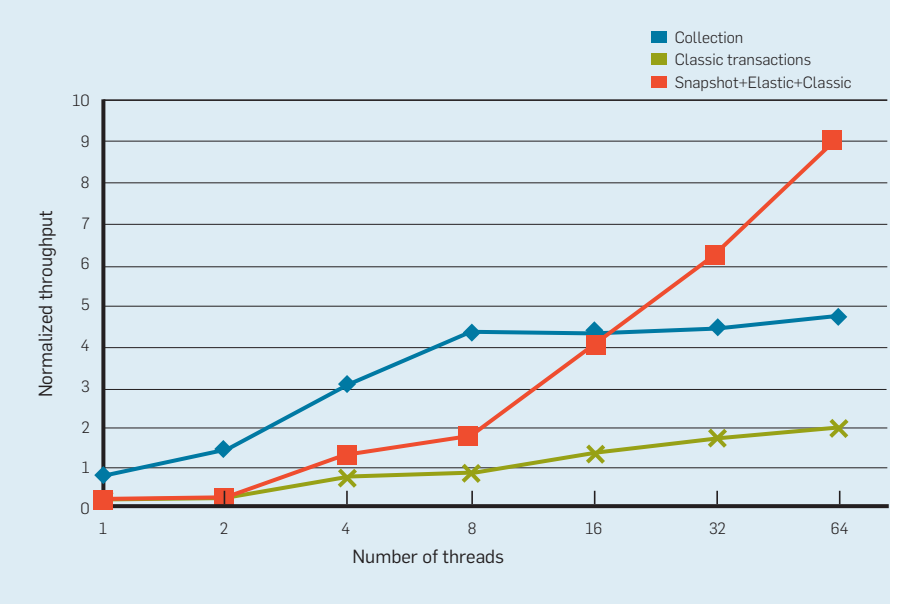
However, the mixture of the snapshot with classic and elastic transactions requires the transaction system make sure all updates (elastic and classic) record the old value before overriding it.

The mixture problem might be more subtle if a relaxed transaction ignores a conflict involving a concurrent strong transaction that cannot ignore it. Elastic and opaque transactions typically handle this issue for read-write conflicts by requiring only the reading transaction decides on conflict resolution. Unlike writes, reads are idempotent so the semantics of the writing transaction is never altered by the outcome of the conflict resolution. Our solution relies on two features: having invisible reads, so the writing transaction does not observe the conflict, and enforcing commit-time validation, so the reading transaction always detects the conflict.

A consequent algorithmic challenge relates to the composition of the semantics. Bob can directly nest Alice's elastic transactions into another transaction, choosing to label it as elastic, snapshot, or classic, guaranteeing atomicity and deadlock freedom of its own operation; for example, one can imagine Alice provides an elastic `contains`(x) Bob composes into a snapshot `containsAll`(C) method that returns successfully only if all elements of a collection C are present. For safety's sake, the strongest semantics of the related transactions (in this case the snapshot transaction) applies to all methods. Hence, a novice programmer, unaware of the various semantics, will always obtain a safe composite transactional method whose opacity would be conveyed to inner transactions. Which semantics to apply (when the semantics are incomparable) is an open question.

Effect on performance. To investigate the potential benefit of mixing transactions of different semantics, we ran the mixed transactions on the collection benchmarks in the exact same settings as before and reported both the new and the previously obtained results (see Figure 5). Each of the three parse operations—`contains`, `add`, and `remove`—is imple-

Figure 5. Throughput (normalized over the sequential throughput) of mixed transactions, classic transactions, and a collection package.



mented through an elastic transaction, and the `size` operation, which returns an atomic snapshot of the number of elements, is implemented through a snapshot transaction. The mixed transaction model performs 4.3x faster than the classic transaction model, TL2, improving on the concurrent collection package by 1.9x on 64 threads. Due to snapshot semantics, the size operation commits more frequently than with a classic transaction. The reason is a snapshot size could return values that were concurrently overridden, where classic size would be aborted. Even though the overhead of polymorphic transactions makes them slower than the concurrent collection package at low levels of parallelism, the performance scales well, compensating for the overhead effect at high levels of parallelism.

The mixture of elastic and classic transactions has been shown to be effective in a non-managed language—C/C++—as well. It improved the performance of the tree library implemented in the transactional vacation-reservation benchmark by 15%;[3] it also improved the performance of a list-based set running on a many-core architecture by about 40x.[9]

Conclusion

The transaction is a proven, appealing abstraction that has been the main topic of many practical and theoretical achievements in research, despite never being widely adopted in practice. The reason the transaction abstraction is appealing as a programming construct is also the reason it might not be used in practice. That is, the appeal of transactions comes from their simplicity and bringing multicore programming to novice programmers. Average programmers can write concurrent code and, with little effort, use transactions to protect shared data against incorrectness. However, the simplicity of the concept is also its main source of rigidity, preventing expert programmers from exploiting their skills and enabling as much concurrency as they could, thereby limiting performance scalability. This limitation is inherent to the concept, not simply a matter of use.

Here, we have suggested a way out by truly democratizing the transaction concept and promoting the coexistence of different transactional semantics in the same application. Although novice programmers would still be able to exploit the simplicity of the transaction abstraction in its original (strong and hence simple) form, expert programmers would be able to exploit, whenever possible, more expressive semantics of relaxed transaction models to gain in concurrency.

As this polymorphism helps expert programmers take full advantage of transactions, they can likewise develop new efficient libraries that motivate other programmers to adopt this abstraction. It also raises new challenges for guaranteeing the various semantics can be used effectively in the same system. C

References

1. Almes, G.T., Black, A.P., Lazowska, E.D., and Noe, J.D. The Eden system: A technical review. *IEEE Transactions on Software Engineering 11*, 1 (Jan. 1985), 43–59.
2. Cascaval, C., Blundell, C., Michael, M., Cain, H.W., Wu, P., Chiras, S., and Chatterjee, S. Software transactional memory: Why is it only a research toy? *Queue 6*, 5 (Sept. 2008), 46–58.
3. Crain, T., Gramoli, V., and Raynal, M. A speculation-friendly binary search tree. In *Proceedings of ACM SIGPLAN Conference on the Principles and Practice of Parallel Computing* (New Orleans, Feb. 23–27). ACM Press, New York, 2012, 161–170.
4. Dice, D., Lev, Y., Moir, M., and Nussbaum, D. Early experience with a commercial hardware transactional memory implementation. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems* (Washington, D.C., Mar. 7–11). ACM Press, New York, 2009, 157–168.
5. Dice, D., Shalev, O., and Shavit, N. Transactional locking II. In *Proceedings of the International Symposium on DIStributed Computing, Vol. 4167 of LNCS* (Stockholm, Sept. 18–20). Springer, 2006, 194–208.
6. Dragojevic, A., Felber, P., Gramoli, V., and Guerraoui, R. Why STM can be more than a research toy. *Commun. ACM 54*, 4 (Apr. 2011), 70–77.
7. Eswaran, K.P., Gray, J.N., Lorie, R.A., and Traiger, I.L. The notions of consistency and predicate locks in a database system. *Commun. ACM 19*, 11 (Nov. 1976), 624–633.
8. Felber, P., Gramoli, V., and Guerraoui, R. Elastic transactions. In *Proceedings of the International Symposium on DIStributed Computing, Vol. 5805 of LNCS* (Elche, Spain, Sept. 23–25). Springer, 2009, 93–107.
9. Gramoli, V., Guerraoui, R., and Trigonakis, V. TM2C: A software transactional memory for many-cores. In *Proceedings of EuroSys* (Bern, Switzerland, Apr. 10–13). ACM Press, New York, 2012, 351–364.
10. Gramoli, V., Harmanci, D., and Felber, P. On the input acceptance of transactional memory. *Parallel Processing Letters 20*, 1 (Mar. 2010), 31–50.
11. Greenwald, M. Two-handed emulation: How to build non-blocking implementations of complex data-structures using DCAS. In *Proceedings of the Symposium on Principles of Distributed Computing* (Monterey, CA, July 21–24). ACM Press, New York, 2002, 260–269.
12. Guerraoui, R., Capobianchi, R., Lanusse, A., and Roux, P. Nesting actions through asynchronous message passing: The ACS protocol. In *Proceedings of the European Conference on Object-Oriented Programming, Vol. 615 of LNCS* (Utrecht, The Netherlands, June 29–July 3). Springer, 1992, 170–184.
13. Guerraoui, R. and Kapalka, M. *Principles of Transactional Memory*. Morgan & Claypool, San Rafael, CA, 2010.
14. Harris, T. A pragmatic implementation of non-blocking linked-lists. In *Proceedings of the International Symposium on DIStributed Computing, Vol. 2180 of LNCS* (Lisboa, Portugal, Oct 3–5). Springer, 2001, 300–314.
15. Harris, T., Marlow, S., Peyton-Jones, S., and Herlihy, M. Composable memory transactions. In *Proceedings of the ACM SIGPLAN Conference on the Principles and Practice of Parallel Computing* (Chicago, June 15–17). ACM Press, New York, 2005, 48–60.
16. Herlihy, M. and Koskinen, E. Transactional boosting: A methodology for highly concurrent transactional objects. In *Proceedings of ACM SIGPLAN Conference on the Principles and Practice of Parallel Computing* (Salt Lake City, Feb. 20–23). ACM Press, New York, 2008, 207–246.
17. Herlihy, M., Luchangco, V., Moir, M., and Scherer III, W.N. Software transactional memory for dynamic-sized data structures. In *Proceedings of the ACM Symposium on the Principles of Distributed Computing* (Boston, July 13–16). ACM Press, New York, 2003, 92–101.
18. Herlihy, M. and Moss, J.E.B. Transactional memory: Architectural support for lock-free data structures. *SIGARCH Computer Architecture News 21*, 2 (May 1993), 289–300.
19. Knight, T. An architecture for mostly functional languages. In *Proceedings of the ACM Conference on LISP and Functional Programming* (Cambridge, MA, Aug. 4–6, 1986), 105–112.
20. Korland, G., Shavit, N., and Felber, P. Deuce: Noninvasive software transactional memory. *Transactions on High-Performance Embedded Architectures and Compilers 5*, 2 (Jan. 2010).
21. Liskov, B. The Argus language and system. In *Proceedings of Distributed Systems: Methods and Tools for Specification, An Advanced Course, Vol. 190 of LNCS*. Springer, 1985, 343–430.
22. Liskov, B. and Scheifler, R. Guardians and actions: Linguistic support for robust, distributed programs. In *Proceedings of the Symposium on the Principles of Programming Languages* (Albuquerque, NM). ACM Press, New York, 1982, 7–19.
23. Lynch, N.A. Multilevel atomicity a new correctness criterion for database concurrency control. *ACM Transactions on Database Systems 8*, 4 (Dec. 1983), 484–502.
24. Moss, J.E.B. Open nested transactions: Semantics and support. Poster presentation at the Workshop on Memory Performance (Austin, TX, 2006).
25. Papadimitriou, C.H. The serializability of concurrent database updates. *Journal of the ACM 26*, 4 (Oct. 1979), 631–653.
26. Ramadan, H.E., Roy, I., Herlihy, M., and Witchel, E. Committing conflicting transactions in an STM. In *Proceedings of the ACM SIGPLAN Conference on the Principles and Practice of Parallel Computing* (Raleigh, NC, Feb. 14–18), ACM Press, New York, 2009, 163–172.
27. Reuter, A. Concurrency on high-traffic data elements. In *Proceedings of the ACM Conference on the Principles of Database Systems* (Los Angeles, Mar. 29–31). ACM Press, New York, 1982, 83–92.
28. Scherer III, W.N. and Scott, M.L. Advanced contention management for dynamic software transactional memory. In *Proceedings of the ACM Symposium on the Principles of Distributed Computing* (Las Vegas, July 17–25). ACM Press, New York, 2005, 240–248.
29. Shavit, N. and Touitou, D. Software transactional memory. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (Ottawa, Aug. 20–23). ACM Press, New York, 1995, 204–213.
30. Shpeisman, T., Menon, V., Adl-Tabatabai, A.-L., Balensiefer, S., Grossman, D., Hudson, R.L., Moore, K.F., and Saha, B. Enforcing isolation and ordering in STM. In *Proceedings of the ACM Conference on Programming Language Design and Implementation* (San Diego, CA). ACM Press, New York, 2007, 78–88.

Vincent Gramoli (vincent.gramoli@sydney.edu.au) is an assistant professor at the University of Sydney and a researcher at the National Information and Communication Technology Australia (NICTA).

Rachid Guerraoui (rachid.guerraoui@epfl.ch) is a professor at École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

DOI:10.1145/2500887

What do we know now that we did not know 40 years ago?

BY XUEDONG HUANG, JAMES BAKER, AND RAJ REDDY

A Historical Perspective of Speech Recognition

WITH THE INTRODUCTION of Apple's Siri and similar voice search services from Google and Microsoft, it is natural to wonder why it has taken so long for voice recognition technology to advance to this level. Also, we wonder, when can we expect to hear a more human-level performance? In 1976, one of the authors (Reddy) wrote a comprehensive review of the state of the art of voice recognition at that time. A non-expert in the field may benefit from reading the original article.[34] Here, we provide our collective historical perspective on the advances in the field of speech recognition. Given the space limitations, this article will not attempt a comprehensive technical review, but limit the scope to discussing the missing science of speech recognition 40 years ago and what advances seem to have contributed to overcoming some of the most thorny problems.

» key insights

- The insights gained from the speech recognition advances over the past 40 years are explored, originating from generations of Carnegie Mellon University's R&D.
- Several major achievements over the years have proven to work well in practice for leading industry speech recognition systems from Apple to Microsoft.
- Speech recognition will pass the Turing Test and bring the vision of Star Trek-like mobile devices to reality. It will help to bridge the gap between humans and machines. It will facilitate and enhance natural conservation among people. Six challenges need to be addressed before we can realize this audacious dream.

IMAGES FROM SHUTTERSTOCK.COM

Speech recognition had been a staple of science fiction for years, but in 1976 the real-world capabilities bore little resemblance to the far-fetched capabilities in the fictional realm. Nonetheless, Reddy boldly predicted it would be possible to build a $20,000 connected speech system within the next 10 years. Although it took longer than projected, not only were the goals eventually met, but the system costs were much less and have continued to drop dramatically. Today, in many smartphones, the industry delivers free speech recognition that significantly exceeds Reddy's speculations. In most fields the imagination of science fiction writers far exceeds reality. Speech recognition is one of the few exceptions. Moreover, speech recognition is unique not just because of its successes: in spite of all the accomplishments, additional challenges remain that are as daunting as those that have been overcome to date.

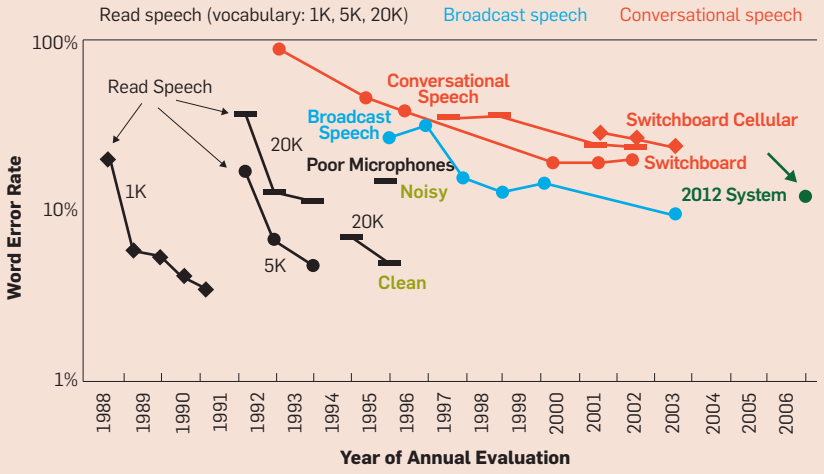
In 1995, Microsoft SAPI was first shipped in Windows 95 to enable application developers to create speech applications on Windows. In 1999 the VoiceXML forum was created to support telephony IVR. While speech-enabled telephony IVR was commercially successful, it has been shown the "speech in" and "screen out" multimodal metaphor is more natural for information consumption. In 2001, Bill Gates demonstrated such a prototype codenamed MiPad at CES.[16] MiPad illustrated a vision on speech-enabled multimodal mobile devices. With the recent adoption of speech recognition used in Apple, Google, and Microsoft products, we are witnessing the ever-improved ability of devices to handle relatively unrestricted multimodal dialogues. We see the fruits of several decades of R&D in spite of remaining challenges. We believe the speech community is en route to pass the Turing Test in the next 40 years with the ultimate goal to match and exceed a human's speech recognition capability for everyday scenarios.

Here, we highlight major speech recognition technologies that worked well in practice and summarize six challenging areas that are critical to move speech recognition to the next level from the current showcase services on mobile devices. More comprehensive technical discussions may be found in the numerous technical papers published over the last decade, including *IEEE Transactions on Audio, Speech and Language Processing* and *Computer Speech and Language*, as well as proceedings from ICASSP, Interspeech, and IEEE workshops on ASRU. There are also numerous articles and books that cover systems and technologies developed over the last four decades.[9,14,15,19,25,33,36,43]

Basic Speech Recognition

In 1971, a speech recognition study group chaired by Allen Newell recommended that many more sources of knowledge be brought to bear on the problem. The report discussed six levels of knowledge: acoustic, parametric, phonemic, lexical, sentence, and semantic. Klatt[23] provides a review of performance of various ARPA-funded speech understanding systems initiated to achieve the goals of Newell report.

Figure 1. Historical progress of speech recognition word error rate on more and more difficult tasks.[10] The latest system for the switchboard task is marked with the green dot.



What we did not know how to do in 1976.v

Statistical modeling and machine learning: Elaboration of HMM, context-dependent phoneme modeling, statistical smoothing and back-off strategies, DNN, semi-supervised learning, discriminative training such as Maximum Mutual Information Estimation (MMIE) and MPE

Training data and computing resources: Several orders of magnitude increase in the size of speech (thousands of hours) and text data (trillions of words) accompanied by the steadily increased distributed CPU and RAM resources

Signal processing dealing with noisy environments: DNN-learned features, MFCC appropriate for Gaussian mixture models, lower-level raw features such as filterbanks appropriate for DNN, Cepstral mean subtraction, 1st and 2nd order delta features, online environment adaptation, and noise-canceling microphone/microphone array

Vocabulary size and dis-fluent speech: From thousands to millions of words supported by n-grams and RNN as the language model, explicit garbage models, and the flexibility to add new words with grapheme form

Speaker independent and adaptive speech recognition: Mixture distributions, speaker training data across different dialects and populations, vocal tract normalization, Maximum a Posteriori (MAP), Maximum Likelihood Linear Regression (MLLR), and unsupervised speaker-adaptive learning

Efficient decoder: Time-synchronous Viterbi search and A* stack decoder with sophisticated pruning techniques, distributed implementation to support large-scale server-based runtime decoder

Spoken language understanding and dialog: Case-frame based robust parser, semi-Markov conditional random field (CRF), boosted decision tree, rule-based or Markov decision process-based dialog management, and recurrent neural networks for sentence understanding

By 1976, Reddy was leading a group at Carnegie Mellon University that was one of a small number of research groups funded to explore the ideas in the Newell report under a multiyear Defense Advanced Research Project Agency (DARPA)-sponsored Speech Understanding Research (SUR) project. This group developed a sequence of speech recognition systems: Hearsay, Dragon, Harpy, and Sphinx I/II. Over a span of four decades, Reddy and his colleagues created several historic demonstrations of spoken language systems, for example, voice control of a robot, large-vocabulary connected-speech recognition, speaker-independent speech recognition, and unrestricted vocabulary dictation. Hearsay-I was one of the first systems capable of continuous speech recognition. The Dragon system was one of the first systems to model speech as a hidden stochastic process. The Harpy system introduced the concept of Beam Search, which for decades has been the most widely used technique for efficient searching and matching. Sphinx-I, developed in 1987, was the first system to demonstrate speaker-independent speech recognition. Sphinx-II, developed in 1992, benefited largely from tied parameters to balance trainability and efficiency at both Gaussian mixture and Markov state level, which achieved the highest recognition accuracy in DARPA-funded speech benchmark evaluation in 1992.

As per the DARPA-funded speech evaluations, the speech recognition word error rate has been used as the main metric to evaluate the progress. The historical progress also directed the community to work on more difficult speech recognition tasks as shown in Figure 1. On the latest switchboard task, the word error rate is approaching an impressive new milestone by both Microsoft and IBM researchers respectively,[4,22,37] following the deep learning framework pioneered by researchers at the University of Toronto and Microsoft.[5,14]

It was anticipated in the early 1970s that to bring to bear the higher-level sources of knowledge might require significant breakthroughs in artificial intelligence. The architecture of the Hearsay system was designed so that many semiautonomous modules can communicate and cooperate in

a speech recognition task while each concentrated on its own area of expertise. In contrast, the Dragon, Harpy, and Sphinx I/II systems were all based on a single, relatively simple modeling principle of joint global optimization. Each of the levels in the Newell report was represented by a stochastic process known as a hidden Markov process. Successive levels were conceptually embedded like nesting blocks, so the combined process was also a (very large) hidden Markov process.[2]

The decoding process of finding the best matched word sequence W to match input speech X is more than a simple pattern recognition problem, since one faces a practically astronomical number of word patterns to search. The decoding process in a speech recognizer's operation is to find a sequence of words whose corresponding acoustic and language models best match the input feature vector sequence. Thus, such a decoding process with trained acoustic and language models is often referred to as a search process. Graph search algorithms, which have been explored extensively in the fields of artificial intelligence, operations research, and game theory, serve as the basic foundation for the search problem in speech recognition.

The importance of the decoding process is best illustrated by Dragon NaturallySpeaking, a product that took 15 years to develop under the leadership of one of the authors (Baker). It has survived for 15 years through many generations of computer technology after being acquired by Nuance. Dragon Systems did not owe its success to inventing radically new algorithms with superior performance. The development of technology for Dragon NaturallySpeaking may be compared with the general development in the same timeframe reviewed in this article. The most salient difference is not algorithms with a lower error rate, but rather an emphasis on simplified algorithms with a better cost-performance trade-off. From its founding, the long-term goal of Dragon Systems was the development of a real-time, large-vocabulary, continuous-speech dictation system. Toward that end, Dragon formulated a coherent mission statement that would last for decades that would be required to reach the long-term goal, but that in each time frame would translate into appropriate short-term and medium-term objectives: Produce the best speech recognition that could run in real time on the current generation of desktop computers.

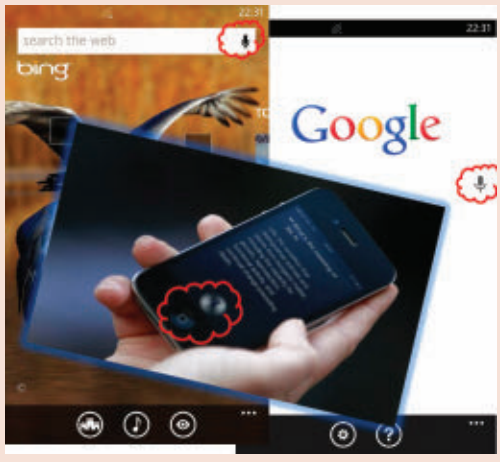
What We Did Not Know in 1976

Each of the components illustrated in Reddy's original review paper has made significant progress. We do not plan to enumerate all the different systems and approaches developed over the decades. Table 1 contains the major achievements that are proven to work well in practice for leading industry speech recognition systems. Today, we can use open research tools, such as HTK, Sphinx, Kaldi, CMU LM toolkit, and SRILM to build a working system. However, the competitive edge in the industry mostly benefited from using a massive amount of data available in the cloud to continuously update and improve the acoustic model and the language model. Here, we discuss progress that enabled today's voice search on mobile phones such as Apple, Google, and Microsoft Voice Search as illustrated in Figure 2.

The establishment of the statistical machine-learning framework, supported by the availability of computing infrastructure and massive training data, constitutes the most significant driving force in advancing the development of speech recognition. This enabled machine learning to treat phonetic, word, syntactic, and semantic knowledge representations in a unified manner. For example, explicit segmentation and labeling of phonetic strings is no longer necessary. Phonetic matching and word verification are unified with word sequence generation that depends on the highest overall rating typically using a context-dependent phonetic acoustic model.

Statistical machine learning. Early methods of speech recognition aimed to find the closest matching sound label from a discrete set of labels. In non-probabilistic models, there is an estimated "distance" between sound labels based on how similar two sounds are estimated to be. In one form, probability models use an estimate of the conditional probability of observing a particular sound label as the best matching label, conditional on the correct label being the hypothesized label, which is also called the "confusion" probability. To estimate the probability of confusing each possible sound with each possible label requires substantially more training data than estimating the mean of a Gaussian distribution, another common representation. This method corresponds to the "labeling" part of the "segmentation and labeling" described in Reddy's 1976 review, whether accompanied by segmentation or not, as was often done by the 1980s for non-probability-based models. This distance may merely be a

Figure 2. Modern search engines such as Bing and Google both offer a readily accessible microphone button (marked in red) to enable voice search the Web. Apple iPhone Siri, while not a search engine (its Web search is now powered by Bing), has a much larger microphone button for multimodal speech dialogue.



score to be minimized.

A pivotal change in the representation of knowledge in speech recognition was just beginning at the time of Reddy's review paper. This change was exemplified by the representation of speech as a hidden Markov process. This is usually referred to with the acronym HMM for "Hidden Markov Model," which is a slight misnomer because it is the process that is hidden not the model.[2] Mathematically, the model for a hidden Markov process has a learning algorithm with a broadly applicable convergence theorem called the Expectation-Maximization (EM) algorithm.[3,8] In the particular case of a hidden Markov process, it has a very efficient implementation via the Forward-Backward algorithm. Since the late 1980s, statistical discriminative training techniques have also been developed based on maximum mutual information or related minimum error criteria.[1,13,21]

Before 2010, a mixture of HMM-based Gaussian densities have typically been used for state-of-the-art speech recognition. The features for these models are typically Mel-frequency cepstral coefficients (MFCC).[6] While there are many efforts in creating features imitating the human auditory process, we want to highlight one significant development that offers learned feature representation with the introduction of deep neural networks (DNN). Overcoming the inefficiency in data representation by the Gaussian mixture model, DNN can replace the Gaussian mixture model directly.[14] Deep learning can also be used to learn powerful discriminative features for a traditional HMM speech recognition system.[37] The advantage of this hybrid system is that decades of speech recognition technologies developed by speech recognition researchers can be used directly. A combination of DNN and HMM produced significant error reduction[4,14,22,37] in comparison to some of the early efforts.[29,40] In the new system, the speech classes for DNN are typically represented by tied HMM states—a technique directly inherited from earlier speech systems.[18]

Using Markov models to represent language knowledge was controversial. Linguists knew no natural language could be represented even by context-free grammar, much less by a finite state grammar. Similarly, artificial intelligence experts were more doubtful that a model as simple as a Markov process would be useful for representing the higher-level knowledge sources recommended in the Newell report.

However, there is a fundamental difference between assuming that language itself is a Markov process and modeling language as a probabilistic function of a hidden Markov process. The latter model is an approximation method that does not make an assumption about language, but rather provides a prescription to the designer in choosing what to represent in the hidden process. The definitive property of a Markov process is that, given the current state, probabilities of future events will be independent of any additional information about the past history of the process. This property means if there is any information about the past history of the observed process (such as the observed words and sub-word units), then the designer should encode that information with distinct states in the hidden process. It turned out that each of the levels of the Newell hierarchy could be represented as a probabilistic function of a hidden Markov process to a reasonable level of approximation.

For today's state-of-the-art language modeling, most systems still use the statistical *N*-gram language models and the variants, trained with the basic counting or EM-style techniques. These models have proved remarkably powerful and resilient. However, the *N*-gram is a highly simplistic model for realistic human language. In a similar manner with deep learning for significantly improving acoustic modeling quality, recurrent neural networks have also significantly improved the *N*-gram language model.[27] It is worth noting that nothing beats a massive text corpora matching the application domain for most real speech applications.

Training data and computational resources. The availability of speech/text data and computing power has been instrumental in enabling speech recognition researchers to develop and evaluate complex algorithms on sufficiently large tasks. The availability of common speech corpora for speech training, development, and evaluation, has been critical, allowing the creation of complex systems of ever-increasing capabilities. Since speech is a highly variable signal and is characterized by many parameters, large corpora become critical in modeling it well enough for automated systems to achieve proficiency. Over the years, these corpora have been created, annotated, and distributed to the worldwide community by the National Institute of Standard and Technology (NIST), the Linguistic Data Consortium (LDC), European Language Resources Association (ELRA), and other organizations. The character of the recorded speech has progressed from limited, constrained speech materials to huge amounts of progressively more realistic, spontaneous speech.

Moore's Law predicts doubling the amount of computation for a given cost every 12–18 months, as well as a comparably shrinking cost of memory. Moore's Law made it possible for speech recognition to consume the significantly improved computational infrastructure. Cloud-based speech recognition made it more convenient to accumulate an even more massive amount of speech data than ever imagined in 1976. Both Google and Bing indexed the entire Web. Billions of user queries reach the Web search engine monthly. This massive amount of query click data made it possible to create a far more powerful language model for voice search applications.

Signal and feature processing. A vector of acoustic features is computed typically every 10 milliseconds. For each frame a short window of speech data is selected. Typically each window selects about 25 milliseconds of speech, so the windows overlap in time. In 1976, the acoustic features were typically a measure of the magnitude at each of a set of frequencies for each time window, typically computed by a fast Fourier transform or by a filter bank. The magnitude as function of frequency is called the "spectrum" of the short time window of speech, and a sequence of such spectra over time in a speech utterance can be visualized as a spectrogram.[31]

Over the past 30 years or so, modifications of spectrograms led to sig-

nificant improvements in the performance of Gaussian mixture-based HMM systems despite the loss of raw speech information due to such modifications. Deep learning technology aims squarely at minimizing such information loss and at searching for more powerful, deep learning-driven speech representations from raw data. As a result of the success in deep learning, speech recognition researchers are returning to using more basic speech features such as spectrograms and filterbanks for deep learning,[11] allowing the power of machine learning to automatically discover more useful representations from the DNN itself.[37,39]

Vocabulary size. The maximum vocabulary size for large speech recognition has increased substantially since 1976. In fact, for real-time natural language dictation systems in the late 1990s the vocabulary size essentially became unlimited. That is, the user was not aware of which relatively rare words were in the system's dictionary and which were not. The systems tried to recognize every word dictated and counted as an error any word that was not recognized, even if the word was not in the dictionary.

This point of view forced these systems to learn new words on the fly so the system would not keep making the same mistake every time the same word occurred. It was especially important to learn the names of people and places that occurred repeatedly in a particular user's dictation. Significant advances were made in statistical learning techniques for learning from a single example or a small number of examples. The process was made to appear as seamless as possible to the interactive user. However, the problem remains a challenge because modeling new words is still far from seamless when seen from the point of view of the models, where the small-sample models are quite different from the large-data models.

Speaker independent and adaptive systems. Although probability models with statistical machine learning provided a means to model and learn many sources of variability in the speech signal, there was still a significant gap in performance between single-speaker, speaker-dependent models and speaker-independent models intended for the diverse population. Sphinx introduced large vocabulary speaker-independent continuous speech recognition.[24] The key was to use more speech data from a large number of speakers to train the HMM-based system.

Speech recognition is unique not just because of its successes: in spite of all the accomplishments, additional challenges remain that are as daunting as those that have been overcome so far.

Adaptive learning is also applied to accommodate speaker variations and a wide range of variable conditions for the channel, noise, and domain.[24] Effective adaptation technologies enable rapid application integration, and are a key to successful commercial deployment of speech recognition.

Decoding techniques. Architecturally, the most important development in knowledge representation has been searchable unified graph representations that allow multiple sources of knowledge to be incorporated into a common probabilistic framework. The decoding or search strategies have evolved from many systems summarized in Reddy's 1976 paper, such as stack decoding (A* search),[20] time-synchronous beam search,[26] and Weighted Finite State Transducer (WFST) decoder.[28] These practical decoding algorithms made possible large-scale continuous speech recognition.

Non-compositional methods include multiple speech streams, multiple probability estimators, multiple recognition systems combined at the hypothesis level such as ROVER,[12] and multi-pass systems with increased constraints.

Spoken language understanding. Once recognition results are available, it is equally important to extract "meaning" for the recognition results. Spoken language understanding (SLU) mostly relied on case grammars for representing sets of semantic concepts during 1970s. A good example of putting the case grammars for SLU is exemplified by the Air Travel Information System (ATIS) research initiative funded by DARPA.[32,41] In this task, the users can utter queries on flight information in an unrestricted free form. Understanding the spoken language is about extracting task-specific arguments in a given frame-based semantic representation involving frames such as "departure date," and "flight." The slot in these case frames is specific to the domain involved. Finding the value of properties from speech recognition results must be robust to deal with inherent recognition errors as well as a

wide range of different ways of expressing the same concept.

A number of techniques are used to fill frame slots of the application domain from the training data.[30,35,41] Like acoustic and language modeling, deep learning based on recurrent neural networks can also significantly improve filling slots for language understanding.[38]

Six Major Challenges

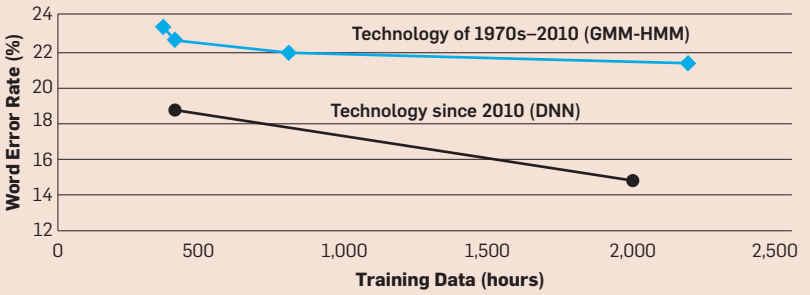
Speech recognition technology is far from perfect. Indeed, technical challenges abound. Based on what we have learned over the past 40 years, we now discuss six of the most challenging areas to be addressed before we can realize the dream of speech recognition.

There is no data like more data. Today we have some very exciting opportunities to collect large amounts of data, thus giving rise to "data deluge." Thanks in large part to the Internet, there are now readily accessible large quantities of everyday speech, reflecting a variety of materials and environments previously unavailable. Recently emerging voice search in mobile phones has provided a rich source of speech data, which, because of the recording of mobile phone users' actions, can be considered as partially "labeled." Apple Siri (powered by Nuance), Google, and Microsoft all have accumulated a massive amount of user data in using voice systems on their products.

New Web-based tools could be made available to collect, annotate, and process substantial quantities of speech in a cost-effective manner in many languages. Mustering the assistance of interested individuals on the Web could generate substantial quantities of language resources very efficiently and cost effectively. This could be especially valuable for creating significant new capabilities for resource "impoverished" languages.

The ever-increasing amount of data presents both an opportunity and a challenge for advancing the state of the art in speech recognition as illustrated in Figure 3, in which our Microsoft colleagues Li Deng and Eric Horvitz used the data from a number of published papers to illustrate the key point. The numbers in Figure 3 are not precise even with our best effort to derive a cohesive chart from data scattered over a period of approximately 10 years.

Figure 3. There is no data like more data. Recognition word error rate vs. the amount of training hours for illustrative purposes only. This figure illustrates how modern speech recognition systems can benefit from increased training data.



We have barely scratched the surface in sampling the many kinds of speech, environments, and channels that people routinely experience. In fact, we currently provide to our automatic systems only a very small fraction of the amount of materials that humans utilize to acquire language. If we want our systems to be more powerful and to understand the nature of speech itself, we need to make more use of it and label more of it. Well-labeled speech corpora have been the cornerstone on which today's systems have been developed and evolved. However, most of the large quantities of data are not labeled or poorly "labeled," and labeling them accurately is costly.

Computing infrastructure. The use of GPUs[5,14] is a significant advancement in recent years that makes the training of modestly sized deep networks practical. A known limitation of the GPU approach is the training speed-up is small when the model does not fit in GPU memory (typically less than six gigabytes). It is recently reported that distributed optimization approach can greatly accelerate deep learning as well as enabling training larger models.[7] A cluster of massive distributed machines has been used to train a modestly sized speech DNN leading to over 10x acceleration in comparison to the GPU implementation.

Moore's Law has been a dependable indicator of the increased capability for computation and storage in our computational systems for decades. The resulting effects on systems for speech recognition and understanding have been enormous, permitting the use of larger and larger training databases and recognition systems, and the incorporation of more detailed models of spoken language. Many of the future research directions and applications implicitly depend upon continued advances in computational capabilities, which seems justified given the recent progress of using distributed computer systems to train large-scale DNNs. With the ever-increased amount of training data as illustrated in Figure 3, it is expected to take weeks or months to train a modern speech system even with a massively distributed computing cluster.

As Intel and others have recently noted, the power density on microprocessors has increased to the point that higher clock rates would begin to melt the silicon. Consequently, industry development is currently focused on implementing microprocessors on multiple cores. The new road maps for the semiconductor industry reflect this trend, and future speed-ups will come more from parallelism than from having faster individual computing elements.

For the most part, algorithm designers for speech systems have ignored investigation of such parallelism, partly because the advancement of scalability has been so reliable. Future research directions and applications will require significantly more computation resources for creating models, and consequently researchers will need to consider massive distributed parallelism in their designs. This will be a significant change from the status quo. In particular, tasks

such as decoding, for which extremely clever schemes to speed up single-processor performance have been developed, will require a complete rethinking of the algorithms. New search methods that explicitly exploit parallelism should be an important research direction.

Unsupervised learning has been successfully used to train a deep network 30-times larger than previously reported.[7] With supervised fine-tuning to get the labels, DNN-based system achieved state-of-the-art performance on ImageNet, a very difficult visual object recognition task. For speech recognition, there is also a practical need to develop high-quality unsupervised or semi-supervised techniques with a massive amount of user interaction data available in the cloud such as click data in the Web search engine.

Upon the successful development of voice search, exploitation of unlabeled or partially labeled data becomes feasible to train the underlying acoustic and language models. We can automatically (and "actively") select parts of the unlabeled data for manual labeling in a way that maximizes its utility. An important reason for unsupervised learning is the systems, like their human "baseline," will have to undergo "lifelong learning," adjusting to evolving vocabulary, channels, language use, among others. There is a need for learning at all levels to cope with changing environments, speakers, pronunciations, dialects, accents, words, meanings, and topics. Like its human counterpart, the system would engage in automatic pattern discovery, active learning, and adaptation.

We must address both the learning of new models and the integration of such models into existing systems. Thus, an important aspect of learning is being able to discern when something has been learned and how to apply the result. Learning from multiple concurrent modalities may also be necessary. For instance, a speech recognition system may encounter a new proper noun in its input speech, and may need to examine textual contexts to determine the spelling of the name appropriately. Success in multimodal unsupervised learning endeavors would extend the lifetime of deployed systems, and directly advance our ability to develop speech systems in new languages and domains without onerous demands of expensive human-labeled data, essentially by creating systems that automatically adapt and improve over time.

Portability and generalizability. An important aspect of learning is generalization. When a small amount of test data is available to adjust speech recognizers, we call such generalization adaptation. Adaptation and generalization capabilities enable rapid speech recognition application integration. There are also attempts to use partially observable Markov decision processes to improve dialogue management if training data can be made available.[42] This set of language resources is often not readily available for many new languages or new tasks. Indeed, obtaining large quantities of training data that is closely matched to the domain is perhaps the single most reliable method to make speech systems work in practice.

Over the past three decades, the speech community has developed and refined an experimental methodology that has helped to foster steady improvements in speech technology. The approach that has worked well is to develop shared corpora, software tools, and guidelines that can be used to reduce differences between experimental setups down to the algorithms, so it becomes easier to quantify fundamental improvements. Typically, these corpora are focused on a particular task. Unfortunately, current language models are not easily portable across different tasks as they lack linguistic sophistication to consistently distinguish meaningful sentences from meaningless ones. Discourse structure is not considered either, merely the local collocation of words.

This strategy is quite different from the human experience. For our entire lives, we are exposed to all kinds of speech data from uncontrolled environments, speakers, and topics, (that is, everyday speech). Despite this variation in our own personal training data we are all able to create internal models of speech and language that are remarkably adept at dealing with variation in the speech chain. This ability to generalize is a key aspect of human speech processing that has not yet found its way into modern speech systems. Research activities on this topic should produce technology that will operate more effectively in novel circumstances, and that can generalize better from smaller amounts of data. Another research area could explore how well information gleaned from large resource languages and/or domains generalize to smaller resource languages and domains.

The challenge here is to create spoken language technologies that are rapidly portable. To prepare for rapid development of such spoken language systems, a new paradigm is needed to study speech and acoustic units that are more language-universal than language-specific phones. Three specific research issues must be addressed: cross-language acoustic modeling of speech and acoustic units for a new target language; cross-lingual lexical modeling of word pronunciations for new language; and cross-lingual language modeling. By exploring correlation between new languages and well-studied languages, we can facilitate rapid portability and generalization. Bootstrapping techniques are keys to building preliminary systems from a small amount of labeled utterances, using them to label more utterance examples in an unsupervised manner, and iterating to improve the systems until they reach a comparable performance level similar to today's high-accuracy systems.

Dealing with uncertainties. The proven statistical DNN-HMM learning framework requires massive amounts of data to deal with uncertainties. How to identify and handle a multitude of variability factors has been key to building successful speech recognition systems. Despite the impressive progress over the past decades, today's speech recognition systems still degrade catastrophically even when the deviations are small in the sense the human listener exhibits little or no difficulty. Robustness of speech recognition remains a major research challenge. We hope for breakthroughs not only in algorithms but also in using the increasingly unsupervised training data available in ways not feasible before.

One pervasive type of variability in the speech signal is the acoustic envi-

ronment. This includes background noise, room reverberation, the channel through which the speech is acquired (such as cellular, Bluetooth, landline, and VoIP), overlapping speech, and Lombard or hyper-articulated speech. The acoustic environment in which the speech is captured and the communication channel through which the speech signal is transmitted represent significant causes of harmful variability that is responsible for drastic degradation of system performance. Existing techniques are able to reduce variability caused by additive noise or linear distortions, as well as compensate for slowly varying linear channels. However, more complex channel distortions such as reverberation or fast-changing noise, as well as the Lombard effect present a significant challenge. While deep learning enabled auto-encoding to create more powerful features, we expect more breakthroughs in learning useful features that may or may not resemble imitating human auditory systems.

Another common type of speech variability studied intensively is due to different speakers' characteristics. It is well known that speech characteristics vary widely among speakers due to many factors, including speaker physiology, speaker style, and accents—both regional and non-native. The primary method currently used for making speech recognition systems more robust is to include a wide range of speakers (and speaking styles) in the training, so as to account for the variations in speaker characteristics. Further, current speech recognition systems assume a pronunciation lexicon that models native speakers of a language and train on large amounts of speech data from various native speakers of the language. Approaches have been explored in modeling accented speech, including explicit modeling of accented speech, adaptation of native acoustic models with only moderate success, as witnessed by some initial difficulties of deploying British English speech system in Scotland. Pronunciation variants have also been incorporated in the lexicon to receive only small gains. Similarly, small progress has been made for detecting speaking rate change.

For the most part, algorithm designers for speech systems have ignored investigation of parallelism, partly because the advance of scalability has been so reliable.

Having Socrates' wisdom. Like most of the ancient Greeks, speech recognition systems lack the wisdom of Socrates. The challenge here is to create systems that reliably detect when they do not know a (correct) word. A clue to the occurrence of such error events is the mismatch between an analysis of a purely sensory signal unencumbered by prior knowledge, such as unconstrained phone recognition, and a word- or phrase-level hypothesis based on higher-level knowledge, often encoded in a language model. A key component of this research would be to develop novel confidence measures and accurate models of uncertainty based on the discrepancy between sensory evidence and a priori beliefs. A natural sequel to detection of such events would be to transcribe them phonetically when the system is confident that its word hypothesis is unreliable, and to devise error-correction schemes.

Current systems have difficulty in handling unexpected—and thus often the most information rich—lexical items. This is especially problematic in speech that contains interjections or foreign or out-of-vocabulary words, and in languages for which there is relatively little data with which to build the system's vocabulary and pronunciation lexicon. A common outcome in this situation is that high-value terms are overconfidently misrecognized as some other common and similar-sounding word. Yet, such spoken events are key to tasks such as spoken term detection and information extraction from speech. Their accurate detection is therefore of vital importance.

Conclusion

Over the last four decades, there have been a number of breakthroughs in speech recognition technologies that have led to the solution of previously impossible tasks. Here, we will summarize the insights gained from the research and product development advances.

In 1976, the computational power available was only adequate to perform speech recognition on highly constrained tasks with low branching factors (perplexity). Today, we are able to handle nearly unlimited vocabularies with much larger branching factors. In 1976, the fastest computer available for routine speech research was a dedi-

cated PDP-10 with 4MB memory. Today's systems have access to a million times more computational power in training the model. Thousands of processors and nearly unlimited collective memory capacity in the cloud are routinely used. These systems can use millions of hours of speech data collected from millions of people from the open population. The power of these systems arises mainly from their ability to collect, process, and learn from very large datasets.

The basic learning and decoding algorithms have not changed substantially in 40 years. However, many algorithmic improvements have been made, such as how to use distributed algorithms for the deep learning task. Surprisingly, even though there is probably enough computational power and memory in iPhone-like smartphone devices, it appears that speech recognition is currently done on remote servers with the results being available within a few hundred milliseconds on the iPhone. This makes it difficult to dynamically adapt to the speaker and the environment, which have the potential to reduce the error rate by half.

Dealing with previously unknown words continues to be a problem for most systems. Collecting very large vocabularies based on Web-based profiling makes it likely that the user would almost always use one of the known words. Today's Web search engines store over 500 million entity entries, which can be powerful to augment the vocabulary that is typically much smaller for speech recognition. The social graph used for Web search engines can also be used to dramatically reduce the needed search space. One final point is that mixed-lingual speech, where phrases from two or more languages may be intermixed, makes the new word problem more difficult.[17] This is often the case for many countries where English is mixed with the native language.

The associated problem of error detection and correction leads to difficult user interface choices for which good enough solutions have been adopted by "Dragon NaturallySpeaking" and subsequent systems. We believe multimodal interactive metaphor will be a dominant metaphor as illustrated by MiPad demo[16] and Apple Siri-like services. We are still missing human-like clarification dialog for new words previously unknown to the system.

Another related problem is the recognition of highly confusable words. Such systems require the use of more powerful discrimination learning. Dynamic sparse data learning, as is routinely done by human beings, is also missing in most of the systems that depend on large data-based statistical techniques.

Speech recognition in the next 40 years will pass the Turing test. It will truly bring the vision of Star Trek-like mobile devices to reality. We expect speech recognition to help bridge the gap between us and machines. It will be a powerful tool to facilitate and enhance natural conservation among people regardless of barriers of location or language, as the *New York Times* story[a] illustrated by Rick Rashid's English to Chinese speech translation demo.[b] C

a http://nyti.ms/190won1

b https://www.youtube.com/watch?v=Nu-nlQqFCKg

References

1. Bahl, L. et al. Maximum mutual information estimation of HMM parameters. In *Proceedings of ICASSP* (1986), 49–52.
2. Baker, J. Stochastic modeling for ASR. *Speech Recognition.* D.R. Reddy, ed. Academic Press, 1975.
3. Baum, L. Statistical Estimation for Probabilistic Functions of a Markov Process. *Inequalities* III, (1972), 1–8.
4. Chen, X., et al. Pipelined back-propagation for context-dependent deep neural networks. In *Proceedings of Interspeech*, 2012.
5. Dahl, G., et al. Context-dependent pre-trained deep neural networks for LVSR. In *IEEE Trans. ASLP 20*, 1 (2012), 30–42.
6. Davis, S. et al. Comparison of parametric representations. *IEEE Trans ASSP 28*, 4 (1980), 357–366.
7. Dean, J. et al. Large scale distributed deep networks. In *Proceedings of NIPS* (Lake Tahoe, NV, 2012).
8. Dempster, et al. Maximum likelihood from incomplete data via the EM algorithm. *JRSS 39*, 1 (1977), 1–38.
9. De Mori, R. Spoken Dialogue with Computers. Academic Press, 1998.
10. Deng, L. and Huang, X. (2004). Challenges in adopting speech recognition. *Commun. ACM 47*, 1 (Jan. 2004), 69–75.
11. Deng, L. et al. Binary coding of speech spectrograms using a deep auto-encoder. In *Proceedings of Interspeech*, 2010.
12. Fiscus, J. Recognizer output voting error reduction (ROVER). In *Proceedings of IEEE ASRU Workshop* (1997), 347–354.
13. He, X., et al. Discriminative learning in sequential pattern recognition. *IEEE Signal Processing 25*, 5 (2008), 14–36.
14. Hinton, G., et al. Deep neural networks for acoustic modeling in SR. *IEEE Signal Processing 29*, 11 (2012).
15. Huang, X., Acero, A., and Hon, H. *Spoken Language Processing.* Prentice Hall, Upper Saddle River, NJ, 2001.
16. Huang, X. et al. MiPad: A multimodal interaction prototype. In *Proceedings of ICASSP* (Salt Lake City, UT, 2001).
17. Huang, J. et al. Cross-language knowledge transfer using multilingual DNN. In *Proceedings of ICASSP* (2013), 7304–7308.
18. Hwang, M., and Huang, X. Shared-distribution HMMs for speech. *IEEE Trans S&AP 1*, 4 (1993), 414–420.
19. Jelinek, F. *Statistical Methods for Speech Recognition.* MIT Press, Cambridge, MA, 1997.
20. Jelinek, F. Continuous speech recognition by statistical methods. In *Proceedings of the IEEE 64*, 4 (1976), 532–557.
21. Katagiri, S. et al. Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method. In *Proceedings of the IEEE 86*, 11 (1998), 2345–2373.
22. Kingsbury, B. et al. Scalable minimum Bayes risk training of deep neural network acoustic models. In *Proceedings of Interspeech* 2012.
23. Klatt, D.H. Review of the ARPA speech understanding project. *JASA 62*, 6 (1977), 1345–1366.
24. Lee, C. and Huo, Q. On adaptive decision rules and decision parameters adaption for ASR. In *Proceedings of the IEEE 88*, 8 (2000), 1241–1269.
25. Lee, K. *ASR: The Development of the Sphinx Recognition System.* Springer-Verlag, 1988.
26. Lowerre, B. The Harpy Speech Recognition System. Ph.D. Thesis (1976). Carnegie Mellon University.
27. Mikolov, T. et al. Extensions of recurrent neural network language model. In *Proceedings of ICASSP* (2011), 5528–5531.
28. Mohri, M. et al. Weighted finite state transducers in speech recognition. *Computer Speech & Language 16* (2002), 69–88.
29. Morgan, N. et al. Continuous speech recognition using mulitlayer perceptions with Hidden Markov Models. In *Proceedings of ICASSP* (1990).
30. Pieraccini R. et al. A speech understanding system based on statistical representation. In *Proceedings of ICASSP* (1992), 193–196.
31. Potter, R., Kopp, G. and Green, H. *Visible Speech.* Van Nostrand, New York, NY, 1947.
32. Price, P. Evaluation of spoken language systems: The ATIS domain. In *Proceedings of the DARPA Workshop*, (Hidden Valley, PA, 1990).
33. Rabiner L. and Juang, B. *Fundamentals of Speech Recognition.* Prentice Hall, Englewood Cliffs, NJ, 1993.
34. Reddy, R. Speech recognition by machine: A review. In *Proceedings of the IEEE 64*, 4 (1976), 501–531; http://www.rr.cs.cmu.edu/sr.pdf.
35. Seneff S. Tina: A NL system for spoken language application. *Computational Linguistics 18*, 1 (1992), 61–86.
36. Tur, G., and De Mori, R. *SLU: Systems for Extracting Semantic Information from Speech.* Wiley, U.K., 2011.
37. Yan, Z., Huo, Q., and Xu, J. A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR. In *Proceedings of Interspeech* (2013).
38. Yao, K. et al. Recurrent neural networks for language understanding. In *Proceedings of Interspeech* (2013), 104–108.
39. Yu, D. et al. Feature learning in DNN—Studies on speech recognition tasks. *ICLR* (2013).
40. Waibel, A. Phone recognition using time-delay neural networks. *IEEE Trans. on ASSP 37*, 3 (1989), 328–339.
41. Ward, W. et al. Recent improvements in the CMU SUS. In *Proceedings of ARPA Human Language Technology* (1994), 213–216.
42. Williams, J. and Young, S. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language 21*, 2 (2007), 393–422.
43. Zue, V. The use of speech knowledge in speech recognition. In *Proceedings of the IEEE 73*, 11 (1985), 1602–1615.

Xuedong Huang is a Distinguished Engineer of Bing Core Search at Microsoft Corp., Redmond, WA, where he founded its Speech Technology Group in 1993. He was previously on the faculty of Carnegie Mellon Univerisity.

James Baker is a former chair, CEO, and co-founder of Dragon Systems in Newton, MA. He received his Ph.D. from Carnegie Mellon University.

Raj Reddy is the Moza Bint Nasser University Professor of Computer Science and Robotics at Carnegie Mellon University in Pittsburgh, PA. He joined CMU in 1969.

Distinguished Speakers Program

talks by and with technology leaders and innovators

Chapters • Colleges and Universities • Corporations • Agencies • Event Planners

A great speaker can make the difference between a good event and a WOW event!

The Association for Computing Machinery (ACM), the world's largest educational and scientific computing society, now provides colleges and universities, corporations, event and conference planners, and agencies – in addition to ACM local Chapters – with direct access to top technology leaders and innovators from nearly every sector of the computing industry.

Book the speaker for your next event through the ACM Distinguished Speakers Program (DSP) and deliver compelling and insightful content to your audience. **ACM will cover the cost of transporation for the speaker to travel to your event.** Our program features renowned thought leaders in academia, industry and government speaking about the most important topics in the computing and IT world today. Our booking process is simple and convenient. Please visit us at: **www.dsp.acm.org**. If you have questions, please send them to **acmdsp@acm.org**.

The ACM Distinguished Speakers Program is an excellent solution for:

Corporations Educate your technical staff, ramp up the knowledge of your team, and give your employees the opportunity to have their questions answered by experts in their field.

Colleges and Universities Expand the knowledge base of your students with exciting lectures and the chance to engage with a computing professional in their desired field of expertise.

Event and Conference Planners Use the ACM DSP to help find compelling speakers for your next conference and reduce your costs in the process.

ACM Local Chapters Boost attendance at your meetings with live talks by DSP speakers and keep your chapter members informed of the latest industry findings.

Captivating Speakers from Exceptional Companies, Colleges and Universities

DSP speakers represent a broad range of companies, colleges and universities, including:

IBM
Microsoft
BBN Technologies
Raytheon

Sony Pictures
McGill University
Tsinghua University
UCLA

Georgia Tech
Carnegie Mellon University
Stanford University
University of Pennsylvania

University of British Columbia
Siemens Information Systems Bangalore
Lawrence Livermore National Laboratory
National Institute of Standards and Technology

Topics for Every Interest

Over 400 lectures are available from 120 different speakers with topics covering:

Software
Cloud and Delivery Methods
Emerging Technologies
Engineering

Web Topics
Computer Systems
Open Source
Game Development

Career-Related Topics
Science and Computing
Artificial Intelligence
Mobile Computing

Computer Graphics, Visualization and Interactive Techniques
High Performance Computing
Human Computer Interaction

Exceptional Quality Is Our Standard

The same ACM you know from our world-class Digital Library, magazines and journals is now putting the affordable and flexible Distinguished Speaker Program within reach of the computing community.

Microsoft Research
The DSP is sponsored in part by Microsoft Europe

Association for Computing Machinery
Advancing Computing as a Science & Profession

DOI:10.1145/2548017

Technical Perspective
Silicon Stress

By Subramanian S. Iyer

MOORE'S LAW, WHICH predicts the doubling of transistor density every two years or so, has been the mainstay of the ubiquitous proliferation of semiconductor electronics and the mobile revolution that has changed our lives for the better since the invention of the transistor by Shockley et al. in 1949 and its application to the integrated circuit by Kilby in the 1950s.

For the past several decades, we have been scaling relentlessly using Dennard's constant electric field scaling method and succeeded in making our chips faster, smaller, and cheaper. Implicit in this evolution is the assumption that we can print these circuits in an economical manner. This assumption is now in question as we have reached dimensions that are significantly below the resolution of the light used to print these features. While we have employed tricks to print these sub-wavelength features, they come at a cost that threatens the expectation of lower cost per circuit. In fact, some project an increased cost per function, which of course begs the question: "Why scale any further?"

Three-dimensional integration (3Di) offers some relief here. It is important to point out that 3Di does not make either a faster transistor or a cheaper transistor per se, but offers the possibility of integrating mature technologies to achieve effective improvements in effective arial transistor density, lower die-to-die latency, and a high degree of componentization by integrating separately optimized and hence less complex technologies. This approach promises to extend the Moore's Law expectation at least for a few more generations.

The TSV allows signals and power to pass through an entire silicon layer and is perhaps the most distinguishing feature of 3D stacking.

There are many embodiments of 3Di, which require the stacking of either partially functional dice or even wafers. A common feature of all these different embodiments is the *Through Silicon Via* (TSV). The TSV literally allows signals and power to pass through an entire silicon layer and is perhaps the most distinguishing feature of 3D stacking. TSVs tend to be rather large compared to the other features in the chip. Moreover, they are lined with thick dielectrics and filled with conductive materials that have a high coefficient of thermal mismatch with respect silicon. The consequence of the introduction of these large dissimilar features is the potential to cause significant stresses in the silicon that can cause structural defects and even failure in the silicon. It is also possible to modulate the electrical properties of the silicon devices, although that may be a less severe effect.

The following paper by Jung et al. is a thorough analysis of the stresses that can result from the introduction of a TSV in silicon. The authors have developed a fairly comprehensive and yet simple method to apply linear superposition to estimate the thermo-mechanical stresses that these TSVs can introduce. The analysis also can be used to estimate a simplified metric called the Von Mises stress that is a measure of mechanical stability. This approach can be used to design stable and reliable TSVs and promises to be a valuable tool in the design of 3D chips.

Subramanian S. Iyer (ssiyer@us.ibm.com) is an IBM Fellow and Director of System Scaling Technology, Microelectronics Division, at the Systems and Technology Group, Hopewell Junction, NY.

DOI:10.1145/2494536

TSV Stress-Aware Full-Chip Mechanical Reliability Analysis and Optimization for 3D IC

By Moongon Jung, Joydeep Mitra, David Z. Pan, and Sung Kyu Lim

Abstract

Three-dimensional integrated circuit (3D IC) with through-silicon-via (TSV) is believed to offer new levels of efficiency, power, performance, and form-factor advantages over the conventional 2D IC. However, 3D IC involves disruptive manufacturing technologies compared to conventional 2D IC. TSVs cause significant thermomechanical stress that may seriously affect performance, leakage, and reliability of circuits. In this paper, we discuss an efficient and accurate full-chip thermomechanical stress and reliability analysis tool as well as a design optimization methodology to alleviate mechanical reliability issues in 3D ICs. First, we analyze detailed thermomechanical stress induced by TSVs in conjunction with various associated structures such as landing pad and dielectric liner. Then, we explore and validate the linear superposition principle of stress tensors and demonstrate the accuracy of this method against detailed finite element analysis (FEA) simulations. Next, we apply this linear superposition method to full-chip stress simulation and a reliability metric named the von Mises yield criterion. Finally, we propose a design optimization methodology to mitigate the mechanical reliability problems in 3D ICs.

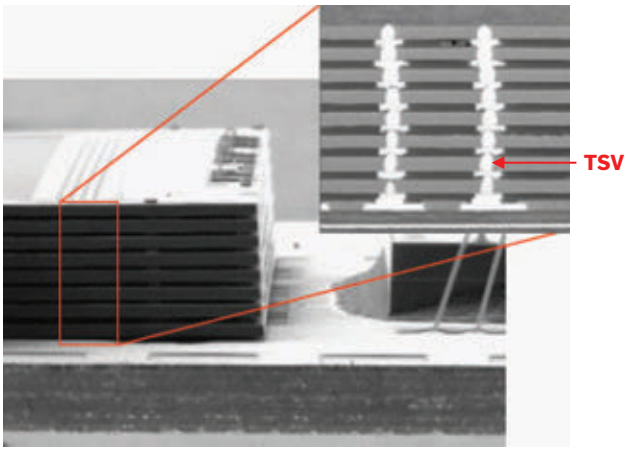
1. WHY 3D IC?

A major focus of the semiconductor industry in the last 4–5 decades has been to miniaturize ICs by advanced lithography patterning technology, which is now around 22 nm node. While ITRS still predicts further CMOS scaling, for example, to around 7 nm node by the year of 2020,[7] such scaling will reach fundamental physical limit, or even before that happens, the economy of scaling will require other means for "more Moore" and "more than Moore" integration.

Due to the increasing power, performance, and financial bottlenecks beyond 32–22 nm, industry began to look for alternative solutions. This has led to the active research, development, and deployment of thinned and stacked 3D ICs, initially by wire-bond, later by flip-chip, and recently by through-silicon-via (TSV).[18]

TSV is the key enabling technology in 3D IC as depicted in Figure 1. This TSV provides vertical signal, power, and thermal paths between the dies in a stack. With 3D integration technology employing TSVs, both the average and maximum distance between components can be substantially reduced by placing them on different dies, which translates into significant savings in delay, power, and area. Moreover, it enables the integration of heterogeneous devices, such as 28 nm for high-speed logic and 130 nm for analog, making the entire system more compact and efficient.

Figure 1. Samsung 16Gb NAND stack (eight 2Gb NAND) with TSV.[20]



Recently, 64 parallel processor cores with stacked memory[12] and a large-scale 3D CMP with a cluster-based near-threshold computing architecture[4] have been demonstrated from academia. Moreover, a heterogeneous 3D FPGA (Xilinx Virtex-7 FPGA) has been already in mass production.[22] However, this new design element, that is, TSV, causes several challenges. The thermomechanical reliability problem caused by TSV-induced stress is one of the biggest challenges in 3D ICs.

2. THERMOMECHANICAL STRESS IN 3D ICs

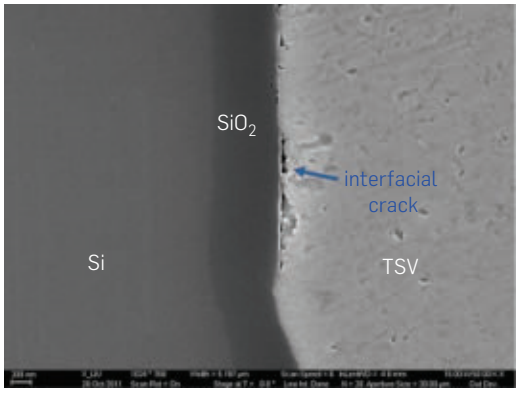
Due to the significant coefficients of thermal expansion (CTE) mismatch between TSV fill material such as copper (= 17 ppm/K) and silicon substrate (= 2.3 ppm/K), thermomechanical stress builds up during 3D IC fabrication process and thermal cycling of TSV structures. Because the copper (= Cu) annealing temperature is much higher than the operating temperature, tensile stress appears on silicon after cooling down to room temperature. This thermomechanical stress can affect both chip performance and reliability.

A previous version of this work was published in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 31*, 8 (2012), 1194–1207.

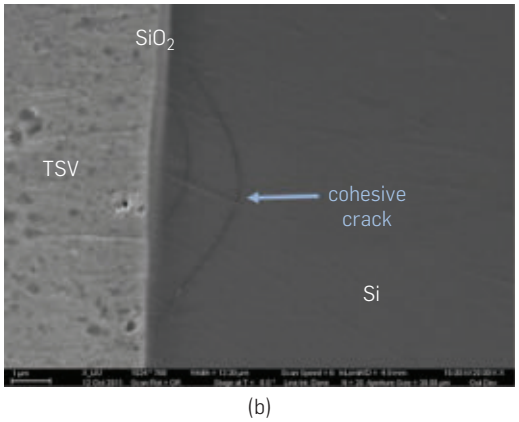
In semiconductors, changes in interatomic spacing resulting from strain affect the bandgaps, making it easier or harder for electrons—depending on the material and strain—to be raised into the conduction band. This results in a change in resistivity of the semiconductor, which can also be translated to a change in mobility.[8] In sub-100 nm nodes, the strained silicon technology has been widely used to boost carrier mobility in the transistor channel. TSV-induced stress affects the carrier mobility on top of this strained silicon and acts as an additional variation source. In fact, the tensile stress induced by TSV affects electron and hole mobility in the opposite direction. Thus, if designers do not take care of this mobility variation in the chip design stage, the intended chip performance is not guaranteed. Previous works[2, 23] discussed the impact of TSV-induced stress on individual device performance as well as full-chip timing.

Meanwhile, there have been major concerns on the thermomechanical reliability of TSV structures. If there is a small defect such as a void around a TSV, TSV-induced stress can drive the interfacial cracking between dielectric liner and silicon substrate or the cohesive cracking in dielectric liner and silicon substrate as shown in Figure 2.[15] These cracks may damage transistors nearby, create conducing paths between TSVs (= short circuit), and cause the entire chip operation failure in the worst case. Previous works studied the crack growth behavior under TSV stress.[9, 14, 19] However, most previous works focused on modeling the thermomechanical stress and reliability of a single TSV in isolation. These simulations were performed using finite element analysis (FEA) methods which are computationally expensive or infeasible for full-chip-scale analysis.

Figure 2. Crack growth due to thermomechanical stress.[15] (a) Interfacial crack between dielectric liner and silicon substrate; (b) cohesive crack in silicon substrate.



(a)



(b)

In this paper, we present a full-chip TSV thermomechanical stress and reliability analysis flow which overcomes the limitation of the FEA method. In addition, we provide a design optimization methodology to reduce mechanical reliability problems in TSV based 3D ICs. To obtain realistic stress distributions across a chip, we first model detailed and practical TSV structures and study their impact on stress, which lacked in many previous works mainly because the design context is not considered. Then, we validate the principle of linear superposition of stress tensors against FEA simulations, and apply this methodology to generate a stress map and a reliability metric map on a full-chip scale. In addition, we present design methods to reduce von Mises stress, which is a mechanical reliability metric that identifies mechanically unstable spots such as crack vulnerable locations, on full-chip 3D IC designs by tuning design parameters such as liner thickness and TSV placement.

3. BASELINE MODELING

3.1. Limitation of existing works

The analytical 2D radial stress model, known as *Lamé* stress solution, was employed to address the TSV thermomechanical stress effect on device performance in Yang et al.[23] This 2D plane solution assumes an infinitely long TSV embedded in an infinite silicon substrate and provides stress distribution in silicon substrate region, which can be expressed as follows[16]:

$$\sigma_{rr}^{Si} = -\sigma_{\theta\theta}^{Si} = -\frac{E\Delta\alpha\,\Delta T}{2}\left(\frac{D_{TSV}}{2r}\right)^2$$
$$\sigma_{zz}^{Si} = \sigma_{rz}^{Si} = \sigma_{\theta z}^{Si} = \sigma_{r\theta}^{Si} = 0 \qquad (1)$$

where σ^{Si} is stress in silicon substrate, E is Young's modulus (= a measure of stiffness of an elastic material), $\Delta\alpha$ is mismatch in CTE, ΔT is differential thermal load, r is the distance from TSV center, and D_{TSV} is TSV diameter.

Even though this closed-form formula is easy to handle, this 2D solution is only applicable to the structure with TSV and substrate only, hence inappropriate for the realistic TSV structure with landing pad and liner. Also, it does not capture the 3D nature of a stress field near the wafer surface around TSVs where devices are located. Moreover, the TSV/substrate interface region near the wafer surface is known to be a highly problematic area for mechanical reliability.[19] In our study, the wafer surface means the silicon surface right below substrate (Si)/dielectric layer (SiO_2) interface.

Though the authors in Ryu et al.[19] proposed a semi-analytic 3D stress model, it is only valid for TSV with a high aspect ratio. Also, their TSV structure only includes TSV and silicon substrate, hence we cannot apply their model to TSV which contains landing pad and dielectric liner because of the change in boundary conditions. Furthermore, since their model is only applicable to a single TSV in isolation, it cannot be directly used to assess mechanical reliability issues in a full-chip scale.

3.2. Our improved structure

Since there is no known analytical stress model for a realistic TSV structure, 3D FEA models for a TSV structure are created to investigate the stress distribution near wafer surface. To realistically examine the thermomechanical stress induced by TSVs, our baseline simulation structure of a TSV is based on the fabricated and the published data,[3, 14] as shown in Figure 3.

We construct two TSV cells, that is, TSV_A and TSV_B, which occupy three and four standard cell rows in NCSU 45 nm technology.[6] We define 1.205 μm and 2.44 μm from TSV edge as keep-out zone (KOZ) in which no cell is allowed to be placed for TSV_A and TSV_B cells, respectively. Our baseline TSV diameter, height, Cu diffusion barrier thickness, liner thickness, and landing pad size are 5 μm, 30 μm, 50 nm, 125 nm, and 6 μm, respectively, unless otherwise specified, which are close to the data in der Plas et al.[3] We use SiO_2 and Ti as a baseline liner and a Cu diffusion barrier material, respectively. Material properties used for our experiments are listed in Table 1. We use the commercial FEA simulation tool ABAQUS to perform experiments, and all materials are assumed to be linear elastic and isotropic. Also, perfect adhesion is assumed at all material interfaces.[17]

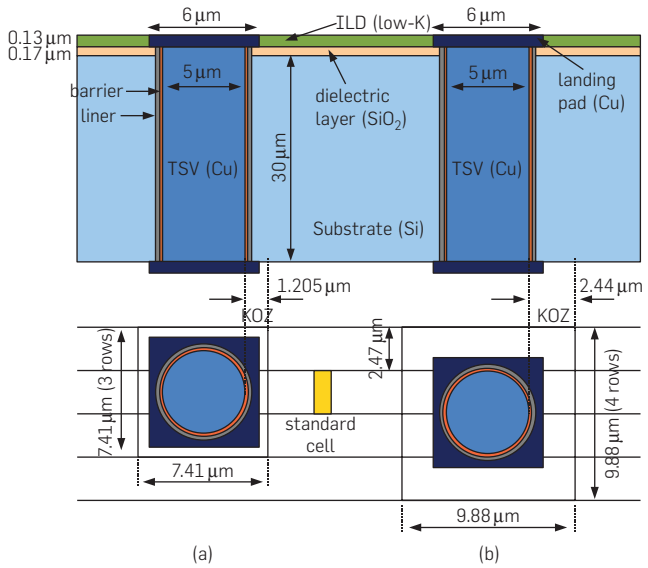
3.3. Stress tensors

Before discussing the detailed stress modeling results, we introduce the concept of a stress tensor. Stress at a point in an object can be defined by the nine-component stress tensor:

$$\sigma = \sigma_{ij} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

where, the first index i indicates that the stress acts on a plane normal to the i axis, and the second index j denotes the direction in which the stress acts. If index i and j are same we call this a normal stress, otherwise a shear stress. Since we adopt a cylindrical coordinate system in this modeling for the cylindrical TSV, index 1, 2, and 3 represent r, θ, and z, respectively.

Figure 3. Baseline TSV structure. (a) TSV_A cell occupying three standard cell rows (KOZ = 1.205 μm). (b) TSV_B cell occupying four standard cell rows (KOZ = 2.44 μm).



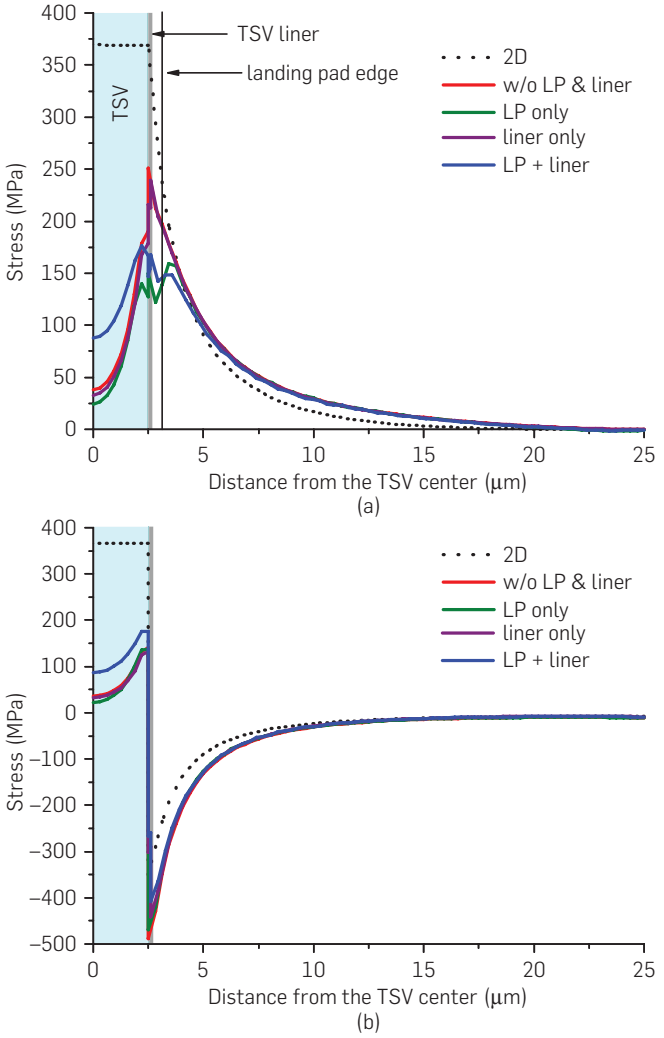
3.4. Stress contours

Figure 4 shows FEA simulation results of a normal stress components σ_{rr} and $\sigma_{\theta\theta}$ along an arbitrary radial line from the TSV center at the wafer surface with $\Delta T = -250$°C of thermal load. That is, we assume TSV structure is annealed at 275°C

Table 1. Material properties.

Material	CTE (ppm/K)	Young's modulus (GPa)	Poisson's ratio
Cu	17	110	0.35
Si	2.3	130	0.28
SiO_2	0.5	71	0.16
Low K	20	9.5	0.3
BCB	40	3	0.34
Ti	8.6	116	0.32
Ta	6.8	186	0.34

Figure 4. Effect of TSV structures on normal stress components. (a) σ_{rr} stress; (b) $\sigma_{\theta\theta}$ stress.



and cooled down to 25°C to mimic the manufacturing process.[11, 16, 19] We also assume that the entire TSV structure is stress free at the annealing temperature.

In our 3D FEA simulations we consider TSV surrounding structures such as dielectric liner and landing pad as well, while the 2D model only considers TSV and substrate which are infinitely long in *z*-direction. Due to this structural difference, we observe the huge discrepancy between 2D solution and 3D stress results at the TSV edge. It is widely known that most of mechanical reliability failures occur at the interface between different materials, hence this TSV edge is the critical region for the reliability. Therefore, the 2D solution does not predict mechanical failure mechanism for TSVs correctly. Also, SiO_2 liner, which acts as a stress buffer layer, reduces σ_{rr} stress at the TSV edge by 35 MPa compared with the case without landing pad and liner. The landing pad also helps decrease stress magnitude at the TSV edge.

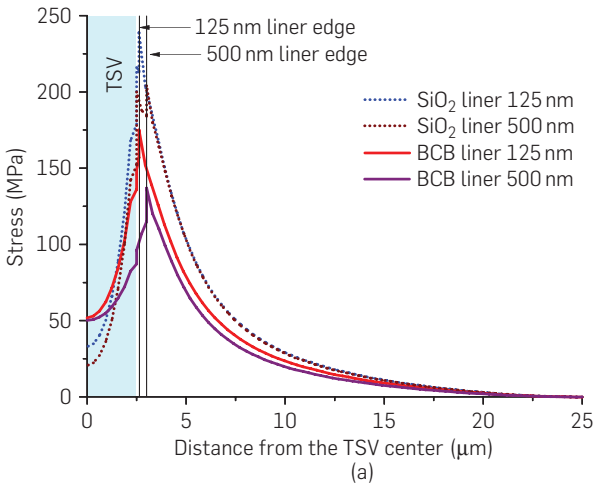
We also employ benzocyclobutene (BCB), a polymer dielectric material, as an alternative TSV liner material.[16, 19] Since Young's modulus, which is a measure of the stiffness of an elastic material, of BCB is much lower than Cu, Si, and SiO_2, this BCB liner can absorb the stress effectively induced by CTE mismatch. Figure 5 shows the impact of liner material and its thickness on σ_{rr} stress component. As liner thickness increases, the stress magnitude at the TSV edge decreases noticeably, especially for the BCB liner case.

It is evident from these simulations that modeling stress distribution considering surrounding structures such as liner and landing pad is important to analyze the thermomechanical stress around TSVs more accurately. We construct a stress library by varying TSV diameter/height, landing pad size, and liner material/thickness to enable full-chip thermomechanical stress and reliability analysis with different TSV structures.

4. FULL-CHIP RELIABILITY ANALYSIS

FEA simulation of thermomechanical stress for multiple TSVs require huge computing resources and time, thus it is not suitable for full-chip analysis. In this section, we present a full-chip thermomechanical stress and reliability analysis flow.

Figure 5. Effect of liner material/thickness on σ_{rr} stress.



To enable a full-chip stress analysis, we first explore the principle of linear superposition of stress tensors from individual TSVs. Based on the linear superposition method, we build full-chip stress map and then compute von Mises yield metric to predict mechanical reliability problems in TSV-based 3D ICs.

4.1. Overview of our full-chip analysis flow

In this section, we briefly describe our full-chip thermomechanical stress and reliability analysis flow. We first perform a detailed FEA simulation of a single TSV and provide the stress tensors along a radial line from the TSV center as an input to our simulation engine. We also provide the locations of the TSVs from 3D IC layout along with a thermal map to the simulation engine. With these inputs, we find a stress influence zone from each TSV. Then, we associate the points in the influence zone with the affecting TSV. Next, for each simulation point under consideration, we look up the stress tensor from the TSV found in the association step, and use the coordinate conversion matrices to obtain stress tensors in the Cartesian coordinate system. We visit an individual TSV affecting this simulation point and add up their stress contributions. Once we finish the stress computation at a point, we calculate the von Mises stress value. The complexity of this algorithm is $O(n)$, where n is number of simulation points.

4.2. Mechanical reliability metric

In order to evaluate if computed stresses indicate possible reliability concerns, a critical value for a potential mechanical failure must be chosen. The von Mises yield criterion is known to be one of the most widely used mechanical reliability metric.[5, 21, 24] If the von Mises stress exceeds a yielding strength, material yielding starts. Prior to the yielding strength, the material will deform elastically and return to its original shape when the applied stress is removed. However, if the von Mises stress exceeds the yield point, some fraction of the deformation will be permanent and nonreversible even if applied stress is removed.

There is a large variation of yield strength of Cu in the literature, from 225 MPa to 600 MPa, and it has been reported to depend upon thickness, grain size, and temperature.[24] We use 600 MPa as a Cu yielding strength in our experiments. The yield strength of silicon is 7000 MPa, which will not be reliability concerns for the von Mises yield criterion.

The von Mises stress is a scalar value at a point that can be computed using components of a stress tensor. By evaluating von Mises stress at the interface between TSV and dielectric liner, where highest von Mises stress occurs, we can predict mechanical failures in TSVs.

4.3. Stress analysis with multiple TSVs

Based on the observation that the stress field of a single TSV in isolation is radially symmetrical due to the cylindrical shape of a TSV, we obtain stress distribution around a TSV from a set of stress tensors along an arbitrary radial line from the TSV center in a cylindrical coordinate system. To evaluate a stress tensor at a point affected by multiple TSVs, a conversion of a stress tensor to a Cartesian coordinate system is required. This is due to the fact that we extract stress tensors

from a TSV whose center is the origin in the cylindrical coordinate system; hence we cannot perform a vector sum of stress tensors at a point from each TSV which has a different center location. That is why we need a universal coordinate system, that is, Cartesian coordinate system in this case.

Then, we compute a stress tensor at the point of interest by adding up stress tensors from TSVs affecting this point. We set a TSV stress influence zone as 25 μm from the center of a TSV with 5 μm diameter, since the magnitude of stress components becomes negligible beyond this distance, which has been verified by FEA simulations.

Let the stress tensor in Cartesian and cylindrical coordinate system be S_{xyz} and $S_{r\theta z}$, respectively.

$$S_{xyz} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}, \; S_{r\theta z} = \begin{bmatrix} \sigma_{rr} & \sigma_{r\theta} & \sigma_{rz} \\ \sigma_{\theta r} & \sigma_{\theta\theta} & \sigma_{\theta z} \\ \sigma_{zr} & \sigma_{z\theta} & \sigma_{zz} \end{bmatrix}$$

The transform matrix Q is the form:

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where, θ is the angle between the X axis and a line from the TSV center to the simulation point. A stress tensor in a cylindrical coordinate system can be converted to a Cartesian coordinate system using conversion matrices: $S_{xyz} = QS_{r\theta z}Q^T$.

4.4. Linear superposition method

A useful principle in the analysis of linearly elastic structures is that of superposition. The principle states that if the displacements at all points in an elastic body are proportional to the forces producing them, the body is linearly elastic. The effect, that is, stresses and displacements, of a number of forces acting simultaneously on such a body is the sum of the effects of the forces applied separately. We apply this principle to compute the stress at a point by adding the individual stress tensors at that point caused by each TSV as follows:

$$S = \sum_{i=1}^{n} S_i$$

where, S is the total stress at the point under consideration and S_i is the individual stress tensor at this point due to the i^{th} TSV.

We validate the linear superposition of stress tensors against FEA simulations by varying the number of TSVs and their arrangement. We set minimum TSV pitch as 10 μm for all test cases. Stress tensors along a radial line from the TSV center in a single TSV structure (stress tensor list) are obtained through FEA simulation with 0.1 μm interval. In our linear superposition method, simulation area is divided into uniform array style grid with 0.05 μm pitch. If the stress tensor at a grid point under consideration is not obtainable directly from the stress tensor list, we compute stress tensor at the point using linear interpolation with adjacent stress tensors in the list.

Table 2 shows some of our comparisons. First, we observe huge run time reduction in our linear superposition method. Note that we perform FEA simulations using 4 CPUs while only one CPU is used for our linear superposition method. Even though our linear superposition method performs stress analysis on a 2D plane at the wafer surface, whereas FEA simulation is performed on entire 3D structure, we can perform stress analysis for other planes in a similar way if needed. Also, run time in our linear superposition method shows linear dependency on the number of simulation points, which is closely related to the number of TSVs under consideration. Thus, our linear superposition method is highly scalable, hence applicable to full-chip scale stress simulations.

Table 2. Von Mises stress comparison between FEA simulations and linear superposition method.

	FEA		Linear superposition		Max % error	
#TSV	#node	Run time	# sim' point	Run time (s)	Inside TSV	Outside TSV
1	153 K	21m 35s	1.0M	20.63	1.0	−0.4
2	282 K	58m 11s	1.2M	26.21	3.3	−0.8
3	358 K	1h 28m 24s	1.44M	36.43	4.8	−1.3
5	546 K	1h 59m 05s	1.68M	56.02	12.7	−1.9
10	1124 K	4h 34m 14s	2.24M	65.32	13.6	−2.0

Most importantly, error between FEA simulations and the linear superposition method is practically negligible. Results show that our linear superposition method overestimates stress magnitude inside TSV. However, though maximum % error inside TSV of 10 TSVs case is as high as 13.6%, stress magnitude difference between FEA and our method is only 5.0 MPa. Also, since most mechanical problems occur at the interface between different materials, this error inside TSV does not pose a serious impact on our reliability analysis. Figure 6 shows the von Mises stress map for one of test cases which contains 10 TSVs, and it clearly shows our linear superposition method matches well with the FEA simulation result.

5. FULL-CHIP SIMULATION RESULTS

We implement a TSV-aware full-chip stress and reliability analysis flow in C++/STL. Four variations of an industrial circuit, with changes in TSV placement style and TSV cell size, are used for our analysis, which are listed in Table 3. The number of TSVs and gates are 1472 and 370K, respectively, for all cases. These circuits are synthesized using Synopsys Design Compiler with the physical library of 45 nm technology,[6] and final layouts are obtained using Cadence SoC Encounter. All circuits are designed to two-die stacked 3D ICs.

We use our in-house 3D placer for TSV and cell placement, and details of TSV and cell placement algorithms can be found in Kim et al.[13] In the regular TSV placement scheme, we pre-place TSVs uniformly on each die, and then place cells, while TSVs and cells are placed simultaneously in the irregular TSV placement scheme. The irregular TSV placement shows better wirelength than the regular case.[13]

5.1. Overall impact study

In this section, we discuss the impact of TSV structure, TSV placement style, and KOZ size on the thermomechanical reliability in 3D ICs. We perform full-chip stress and reliability analysis on our benchmark circuits based on our stress

Figure 6. Sample stress comparison of von Mises stress between FEA simulation and linear superposition method. (a) FEA result; (b) ours; (c) FEA vs. ours along the white line in (a).

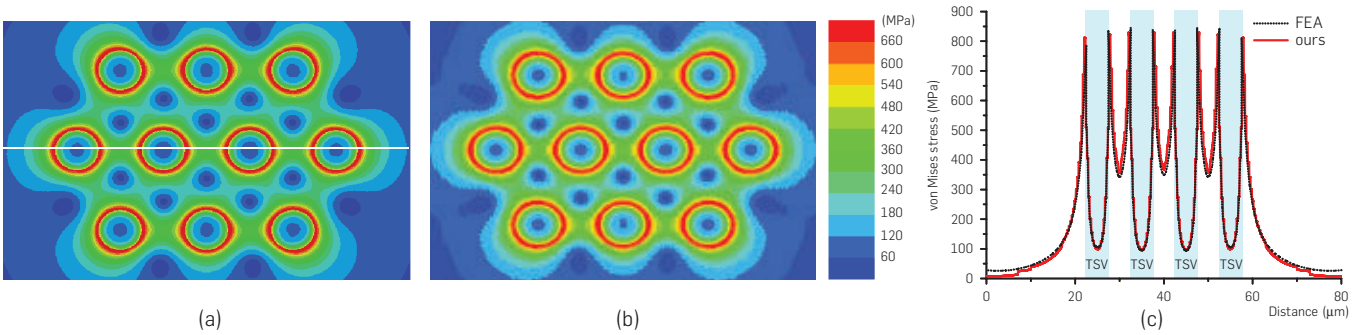


Table 3. Benchmark circuits.

Circuit	TSV placement	TSV cell size (μm × μm)	Wirelength (mm)	Area (μm × μm)
$Irreg_A$	Irregular	7.41 × 7.41	9060	960 × 960
Reg_A	Regular	7.41 × 7.41	9547	960 × 960
$Irreg_B$	Irregular	9.88 × 9.88	8884	1000 × 1000
Reg_B	Regular	9.88 × 9.88	9648	1000 × 1000

modeling results with different TSV structures.

Figure 7 shows the maximum von Mises stress in our benchmark circuits. We first observe that designs with irregular TSV placement show worse maximum von Mises stress than those with the regular TSV placement. This is mainly because TSVs can be placed closely in case of the irregular TSV placement scheme to minimize wirelength. Figure 8 shows the part of von Mises stress maps of $Irreg_A$ and Reg_A circuits, and we see that most of TSVs in the $Irreg_A$ circuit exceed Cu yielding strength (600 MPa).

Second, as the KOZ size becomes larger, stress level reduces significantly for the irregular TSV placement case. By enlarging the KOZ size, that is, increasing TSV cell size in our design flow, TSV pitch increases accordingly. This in turn reduces stress interference between nearby TSVs, and hence decreases von Mises stress level of TSVs. However, for the regular TSV placement case, since the TSV pitch of Reg_A (23.5 μm) and Reg_B (25 μm) are similar and also interference from nearby TSVs is negligible at this distance, there is no noticeable difference in maximum von Mises stress.

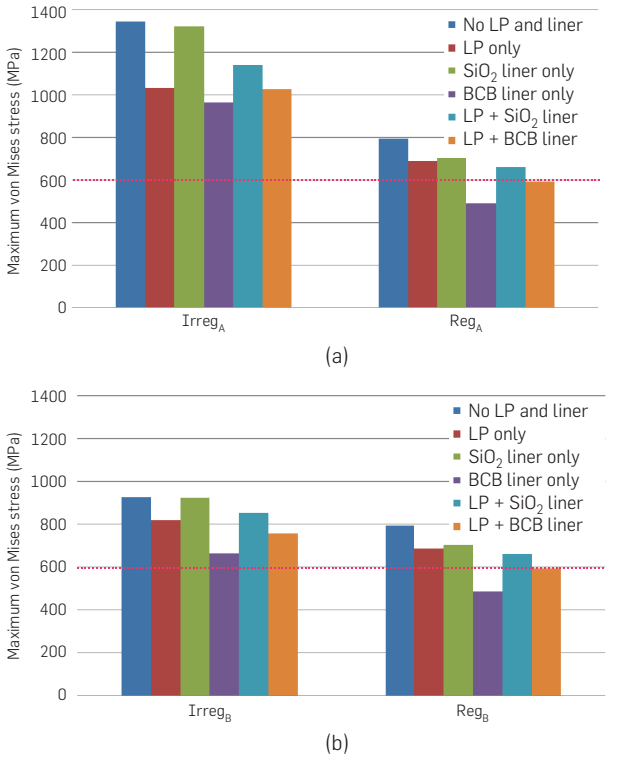
Third, these results show the importance of using an accurate TSV stress model to assess the mechanical reliability of 3D ICs. There are significant differences in the von Mises stress depending on the existence of structures surrounding a TSV, such as landing pad or liner. It is possible that we might overestimate the reliability problems by using a simple TSV stress model not considering landing pad or liner. However, most of these test cases violate the von Mises yield criterion for Cu TSV. Section 5.4 shows how TSV liners help reduce the violations.

5.2. Impact of TSV pitch

TSV pitch is the key factor that determines stress magnitude in the substrate region between TSVs. In this section, we explore the effect of TSV pitch on von Mises stress. We place TSVs regularly on 1 × 1 mm² chip. We use 1600, 2500, 4356, and 10000 TSVs whose pitches are 25, 20, 15, and 10 μm, respectively. We obtain two data sets; one without landing pad, liner, and barrier; and another with 6 × 6 μm² landing pad, 125 nm thick BCB liner, and 50 nm thick Ti barrier.

We first observe that von Mises stress magnitude decreases with increasing pitch and starts to saturate at around 15 μm pitch as shown in Figure 9. This is understandable since the stress magnitude induced by a single TSV becomes negligible at the similar pitch. Also, the layout using TSVs with landing pad and BCB liner shows a similar trend with lower von Mises stress magnitude than the case without these structures.

Figure 7. Impact of TSV structure, TSV placement style, and KOZ size on the maximum von Mises stress. (a) Designs with TSV_A cell (KOZ = 1.205 μm) and (b) those with TSV_B cell (KOZ = 2.44 μm).

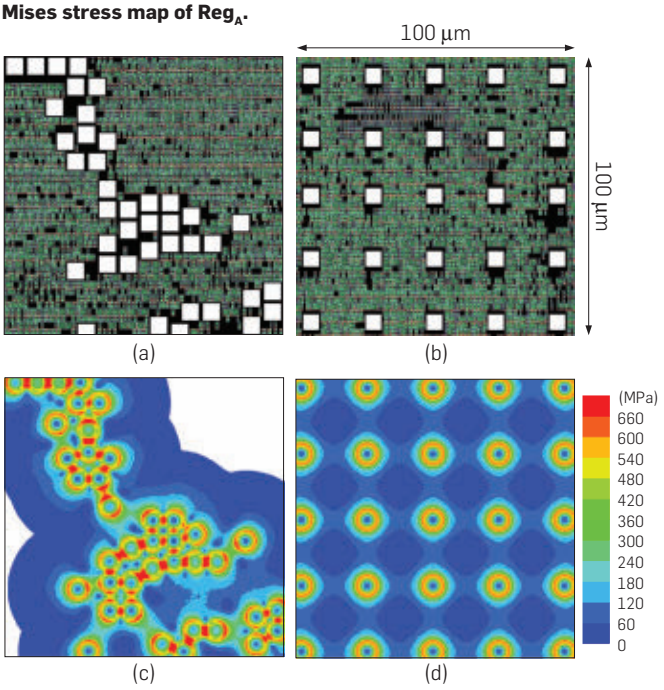


5.3. Impact of TSV dimension

To investigate the effect of the TSV size, we use three different sizes of TSV with a same aspect ratio of 6; TSV small (H/D = 15/2.5 μm and KOZ 1.22 μm), TSV medium (H/D = 30/5 μm and KOZ 1.202 μm), and TSV large (H/D = 60/10 μm and KOZ 1.175 μm), where H/D is TSV height/diameter. Note that these TSV cells are occupying two, three, and five standard cell rows, respectively, which are selected to minimize the KOZ size difference between them. By setting similar KOZ size, we can focus on the impact of TSV size solely. Additionally, we set the landing pad width is 1 μm larger than the corresponding TSV diameter, and use 125 nm thick SiO_2 liner and 50 nm thick Ti barrier for all cases for fair comparisons.

Table 4 shows the maximum von Mises stress. For both irregular and regular TSV placement schemes benefit from smaller TSV diameter significantly. This is mainly because the magnitude of normal stress components decays proportional to $(D/2r)^2$, where r is the distance from the TSV center.

Figure 8. Close-up shots of layouts and von Mises stress maps: (a) $Irreg_A$, (b) Reg_A, (c) von Mises stress map of $Irreg_A$, and (d) von Mises stress map of Reg_A.



5.4. Impact of liner thickness

In this section, we examine the impact of liner thickness on von Mises stress. We use designs with both TSV_A cells and TSV_B cells, and set the landing pad size $6 \times 6\,\mu m^2$ and $8 \times 8\,\mu m^2$, respectively. We also use 50 nm thick Ti barrier for all cases. Figure 10 shows the maximum von Mises stress results with liner thickness of 125 nm, 250 nm, and 500 nm.

We observe that liner thickness has a huge impact on the von Mises stress magnitude, since the thicker liner effectively absorbs thermomechanical stress at the TSV/liner interface. Especially, the BCB liner shows significant reduction in the maximum von Mises stress compared with SiO_2 liner due to extremely low Young's modulus shown in Table 1. For example, 500 nm thick BCB liner reduces the maximum von Mises stress by 29% for the $Irreg_A$ and satisfies the von Mises yield criterion for all circuits with a regular TSV placement.

Table 5 shows the number of TSVs violating von Mises criterion. Even though there are still many TSVs not satisfying von Mises criterion for the $Irreg_A$ circuit, it is possible to reduce von Mises stress if we place TSVs carefully considering this reliability metric during a placement stage.

5.5. Impact of TSV placement optimization

In this section, we manually optimize TSV locations to show the potential benefit of TSV reliability aware layout optimization while minimizing the change in layout. We use the $Irreg_A$ circuit which shows the worst von Mises stress, and employ 500 nm thick BCB liner for this experiment. Our related study with this BCB liner on the maximum von

Table 4. Impact of TSV size on the maximum von Mises stress. The numbers in parentheses are % reduction compared to TSV large case.

	Max von Mises stress (MPa)		
TSV placement	**TSV large**	**TSV medium**	**TSV small**
Irregular	1224.6	1126.4 (8% ↓)	902.7 (26% ↓)
Regular	749.3	654.6 (13% ↓)	449.3 (40% ↓)

Figure 9. Impact of TSV pitch on maximum von Mises stress.

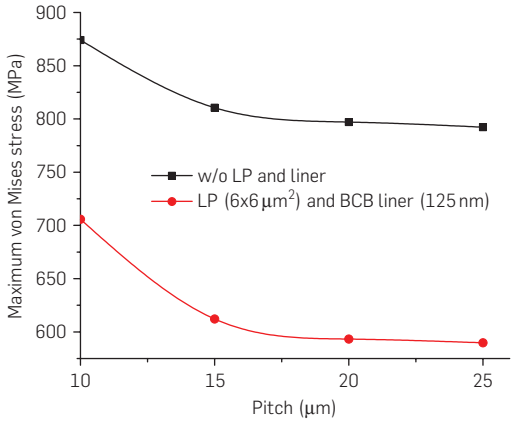
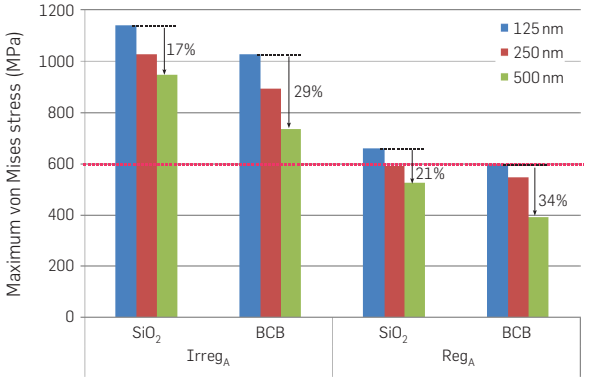


Figure 10. Impact of liner thickness on the maximum von Mises stress of circuits with TSV_A cell.



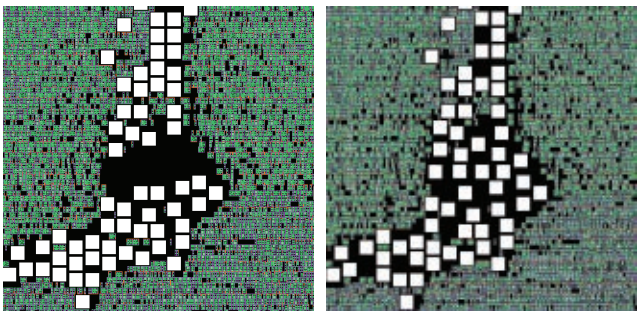
Mises stress vs. TSV-to-TSV pitch shows that 10 μm pitch is a reasonable choice to reduce von Mises stress considering some safety margin. We reposition densely placed TSVs to nearby white spaces if available to reduce the von Mises stress shown in Figure 11.

Table 6 shows the distribution of von Mises stress higher than 480 MPa across the die, wirelength, and longest path delay before and after the TSV replacement. We perform 3D static timing analysis to analyze timing using Synopsys PrimeTime with TSV parasitic information included. We see the reduction in high von Mises stress region after TSV replacement. With small perturbations of TSV locations, we could reduce the von Mises stress level and decrease the number of violating TSVs from 329 to 261, which is 21% improvement with only 0.23% wirelength and 0.81% longest path delay (LPD determines the maximum chip operating frequency) increase, respectively. This small test case shows the possibility of a layout optimization without degrading performance too much.

Table 5. Impact of liner thickness on the number of TSVs violating von Mises criterion. The numbers in parentheses are % reduction compared to the 125 nm thick liner case.

Circuit	Liner material	Violating TSVs 125 nm	250 nm	500 nm
$Irreg_A$	SiO_2	1462	1426 (2% ↓)	1281 (12% ↓)
	BCB	1389	1147 (17% ↓)	329 (76% ↓)
Reg_A	SiO_2	1472	0 (100% ↓)	0 (100% ↓)
	BCB	0	0 (–)	0 (–)
$Irreg_B$	SiO_2	1472	1236 (16% ↓)	64 (96% ↓)
	BCB	974	502 (48% ↓)	0 (100% ↓)
Reg_B	SiO_2	1472	0 (100% ↓)	0 (100% ↓)
	BCB	0	0 (–)	0 (–)

Figure 11. TSV replacement to reduce von Mises stress. TSV landing pads are white rectangles. (a) Original layout; (b) after TSV replacement.



(a) (b)

Table 6. Impact of TSV placement optimization on von Mises stress distribution, wirelength, and longest path delay.

	von Mises stress (MPa) 480–540	540–600	600–660	>660	WL (mm)	LPD (ns)
Orig	0.100%	0.041%	0.011%	0.002%	9060	3.607
Opt	0.092%	0.036%	0.009%	0.0%	9081	3.636

6. CONCLUDING REMARKS

We presented an accurate and fast thermomechanical stress and reliability analysis flow based on the linear superposition principle of stress tensors, which overcomes the limitation of FEA tools, that is, huge computing resources and time. Hence, our method is applicable to large-scale mechanical reliability analysis for TSV-based 3D ICs. Designers can utilize our tool to assess mechanical reliability problems in the 3D IC design and to explore design trade-offs between footprint area, performance, and reliability.

We have worked on a few follow-up studies related to the thermomechanical reliability issues for TSV-based 3D IC. In Jung et al.,[9] we examined the relation between mechanical stress and interfacial crack growth in TSVs. We computed the so called energy release rate (ERR) metric using FEA simulations to measure the probability of a given initial crack in a TSV to grow further. Our studies showed that linear superposition does not hold for ERR calculation for full-chip design. We then employed the response surface model (RSM) method to obtain highly accurate full-chip ERR maps based on our baseline FEA simulations. In Jung et al.,[10] we studied the impact of off-chip elements such as micro-bumps and package bumps on the mechanical reliability of the dies in the 3D stack. Our baseline FEA structure was extended to include these off-chip elements. Related results showed that package bumps lead to a significant background compressive stress to all dies in the stack, which in turn cause the stress contours to shift downward. We developed the so called lateral and vertical linear superposition (LVLS) method to handle stress contributions from off-chip elements in different tiers and obtain full-chip stress map.

Another related study was to investigate how these stress factors (both on-chip and off-chip elements) affect the mobility of the devices nearby and the full-chip timing of 3D ICs.[23] This stress-aware timing information was then used to guide full-chip placement and optimization.[1] Table 1 shows the properties of various materials used in TSV and 3D ICs. Each of these values, however, can vary among TSVs and among the grains inside a single TSV depending on the process technologies used. We are currently looking into the impact of these material property variations on the distribution of mechanical stress tensors, device mobility variations, and full-chip timing and reliability. Lastly, these thermomechanical stress issues are closely related to the electrical reliability of 3D ICs. In Zhao et al.,[25] we examined the impact of TSV stress on electro-migration for power/ground TSVs and the long-term reliability of the power distribution network (PDN) in 3D ICs.

These thermo-electro-mechanical reliability issues in 3D ICs call for holistic multi-physics-based approaches for more effective design solutions. In addition, the industry needs strong collaboration between designers and manufacturers to better tackle these burning issues in TSV and 3D IC and accelerate mainstream acceptance.

Acknowledgments

This work is supported in part by the National Science

Foundation under Grants No. CCF-1018216, CCF-1018750, IBM Faculty Award, and Intel Corporation.

References

1. Athikulwongse, K., Chakraborty, A., Yang, J.S., Pan, D.Z., Lim, S.K. Stress-driven 3D-IC placement with TSV keep-out zone and regularity study. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2010).
2. Athikulwongse, K., Yang, J.S., Pan, D.Z., Lim, S.K. Impact of mechanical stress on the full chip timing for TSV-based 3D ICs. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* (2013).
3. der Plas, G.V. et al. Design issues and considerations for low-cost 3D TSV IC technology. In *IEEE International Solid-State Circuits Conference Digest Technical Papers* (2010).
4. Fick, D., Dreslinski, R., Giridhar, B., Kim, G., Seo, S., Fojtik, M., Satpathy, S., Lee, Y., Kim, D., Liu, N., Wieckowski, M., Chen, G., Mudge, T., Blaauw, D., Sylvester, S. Centip3De: A cluster-based NTC architecture with 64 ARM Cortex-M3 cores in 3D stacked 130 nm CMOS. *IEEE J. Solid-State Circuits 48* (2013).
5. Franssila, S. Introduction to Microfabrication, John Wiley and Sons, 2004.
6. FreePDK45. http://www.eda.ncsu.edu/wiki/FreePDK.
7. International Technology Roadmap for Semiconductors (2012 Update). http://www.itrs.net.
8. Jaeger, R.C., Suhling, J.C., Ramani, R., Bradley, A.T., Xu, J. CMOS stress sensors on (100) silicon. *IEEE J. Solid-State Circuits 35* (2000).
9. Jung, M., Liu, X., Sitaraman, S., Pan, D.Z., Lim, S.K. Full-chip through-silicon-via interfacial crack analysis and optimization for 3D IC. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2011).
10. Jung, M., Pan, D., Lim, S.K. Chip/package co-analysis of thermo-mechanical stress and reliability in TSV-based 3D ICs. In *Proceedings of ACM Design Automation Conference* (2012).
11. Karmarkar, A.P., Xu, X., Moroz, V. Performance and reliability analysis of 3D-integration structures employing through silicon via (TSV). In *IEEE International Reliability Physics Symposium* (2009).
12. Kim, D.H., Athikulwongse, K., Healy, M.B., Hossain, M.M., Jung, M., Khorosh, I., Kumar, G., Lee, Y.J., Lewis, D.L., Lin, T.W., Liu, C., Panth, S., Pathak, M., Ren, M., Shen, G., Song, T., Woo, D.H., Zhao, X., Kim, J., Choi, H., Loh, G.H., Lee, H.H.S., Lim, S.K. 3D-MAPS: 3D massively parallel processor with stacked memory. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers* (2012).
13. Kim, D.H., Athikulwongse, K., Lim, S.K. A study of through-silicon-via impact on the 3D stacked IC layout. In *Proceedings of IEEE International Conference on Computer-Aided Design* (2009).
14. Liu, X., Chen, Q., Dixit, P., Chatterjee, R., Tummala, R.R., Sitaraman, S.K. Failure mechanisms and optimum design for electroplated copper through-silicon vias (TSV). In *IEEE Electronic Components and Technology Conference* (2009).
15. Liu, X., Chen, Q., Sundaram, V., Tummala, R.R., Sitaraman, S.K. Failure analysis of through-silicon vias in free-standing wafer under thermal-shock test. *Microelectronics Reliab. 5* (2013).
16. Lu, K.H., Zhang, X., Ryu, S.K., Im, J., Huang, R., Ho, P.S. Thermo-mechanical reliability of 3-D ICs containing through silicon vias. In *IEEE Electronic Components and Technology Conference* (2009).
17. Ong, J.M.G., Tay, A.A.O., Zhang, X., Kripesh, V., Lim, Y.K., Yeo, D., Chen, K.C., Tan, J.B., Hsia, L.C., Sohn, D.K. Optimization of the thermomechanical reliability of a 65 nm Cu/low-k large-die flip chip package. *IEEE Trans. Compon. Packag. Tech. 32* (2009).
18. Pan, D.Z., Lim, S.K., Athikulwongse, K., Jung, M., Mitra, J., Pak, J., Pathak, M., Seok Yang, J. Design for manufacturability and reliability for TSV-based 3D ICs. In *Proceedings of Asia and South Pacific Design Automation Conference*, (2012).
19. Ryu, S.K., Lu, K.H., Zhang, X., Im, J.H., Ho, P.S., Huang, R. Impact of near-surface thermal stresses on interfacial reliability of through-silicon-vias for 3-D interconnects. In *IEEE Transactions on Device and Material Reliability* (2010).
20. Samsung. 16Gb NAND wafer-level stack with TSV. http://www.samsung.com.
21. Xiang, Y., Chen, X., Vlassak, J.J. The mechanical properties of electroplated Cu thin films measured by means of the bulge test technique. In *Proceedings of Material Research Society Symposium* (2002).
22. Xilinx. Virtex-7 FPGA. http://www.xilinx.com/products/silicon-devices/3dic/index.htm.
23. Yang, J.S., Athikulwongse, K., Lee, Y.J., Lim, S.K., Pan, D.Z. TSV stress aware timing analysis with applications to 3D-IC layout optimization. In *Proceedings of ACM Design Automation Conference* (2010).
24. Zhang, J., Bloomfield, M.O., Lu, J.Q., Gutmann, R.J., Cale, T.S. Modeling thermal stresses in 3-D IC interwafer interconnects. In *IEEE Trans. Semicond. Manuf.* (2006).
25. Zhao, X., Scheuermann, M., Lim, S.K. Analysis of DC current crowding in through-silicon-vias and its impact on power integrity in 3D ICs. In *Proceedings of ACM Design Automation Conference* (2012).

Moongon Jung (moongon@gatech.edu), Georgia Institute of Technology GA.

Joydeep Mitra and David Z. Pan ({joydeep, dpan}@ece.utexas.edu), University of Texas at Austin TX.

Sung Kyu Lim (limsk@ece.gatech.edu), Georgia Institute of Technology GA.

CAREERS

Boise State University
Department of Computer Science
Three Tenure/Tenure-Track Open-Rank Positions

The **Department of Computer Science** at **Boise State University** invites applications for **three tenure/tenure-track open-rank positions**. Applicants should have a commitment to excellence in teaching and a desire to make significant contributions in research by collaborating with faculty and local industry to develop and sustain funded research programs. Senior applicants should have an established track record of research, teaching, and external funding. Preferences will be given to applicants working in the areas of Databases with an emphasis on Big Data, or Human-Computer Interaction with a particular emphasis on usability of user interfaces, or Visualization. An earned Ph.D. in Computer Science or a closely related field is required at the time of appointment.

Boise State has made a significant investment in the growth of the Computer Science department, which is a critical part of the vibrant software and high-tech industry in the Boise metropolitan area. New faculty lines, graduate student support, and a tutoring center have been added to the department. The department is committed to offering a high quality educational experience and in building its research capabilities. For more information, including details on how to apply, please visit us online at http://coen.boisestate.edu/cs/jobs.

Boise State University is strongly committed to achieving excellence through cultural diversity. The University actively encourages applications and nominations of women, persons of color, and members of other underrepresented groups. EEO/AA Institution, Veterans preference may be applicable.

Boston College
Assistant Professor, Computer Science

The Computer Science Department of Boston College invites applications for a tenure-track Assistant Professorship beginning September, 2014. Applications from all areas of Computer Science will be considered. Applicants should have a Ph.D. in Computer Science or related discipline, a strong research record, and a commitment to undergraduate teaching.

We will begin reviewing applications on December 1, 2013, and will continue considering applications until the position is filled. Additional information about the department and the position is available at www.cs.bc.edu. **Submit applications online at apply.interfolio.com/22805**.

Bowling Green State University
Up to two tenure-track positions in Computer Science

We are seeking applicants for tenure-track positions at the **assistant professor** level to teach a variety of courses at the undergraduate and graduate levels and be productive in scholarly research and sponsored projects. Preferred area of specialization is software engineering, including but not limited to: software requirements, software architecture and design, software testing and quality management, software project management, software maintenance, software assurance, design patterns, usability engineering and user interface design. Applicants must hold a Ph.D. or equivalent in CS or a closely related field (or to have completed the requirements for the degree before August 13, 2014), and be committed to excellence in teaching, scholarly research, and obtaining external funding. BGSU offers a small town atmosphere with easy access to Columbus, Detroit, and Ann Arbor. BGSU is an AA/EO employer and encourages applications from women, minorities, veterans, and individuals with disabilities. Email a letter of interest, along with curriculum vitae, statement of teaching philosophy and research agenda, contact information for three professional references, and copies of all transcripts by Sunday, January 12, 2014 to cssearch2014@cs.bgsu.edu. For finalists, three current letters of reference, an official transcript of the highest degree, and a background check are required. For details, go to http://www.bgsu.edu/departments/compsci/jobs.

Bucknell University
Assistant Professor, Computer Science

Applications are invited for a tenure-track position in computer science beginning mid-August 2014. We expect to hire at the Assistant Professor level, but outstanding candidates will be considered at Associate Professor or Professor; years of credit toward tenure will be awarded based upon qualifications. We seek a teacher-scholar with a demonstrated ability to work successfully with a diverse student body and whose research area is in AI/machine learning, algorithms, or human-computer interaction (HCI). 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. Candidates are expected to have completed or be in the final stages of completing their Ph.D. by the beginning of the 2014 fall semester. A strong commitment to excellence in teaching and scholarship is required.

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 accredited. The computing environment is Linux/Unix-based. More information about the department can be found at:

http://www.bucknell.edu/ComputerScience/

Review of applications will begin on January 15 and continue until the position is filled. Candidates are asked to submit a cover letter, CV, 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 Stephen Guattery of the Computer Science Department at guattery@bucknell.edu.

Bucknell University, an Equal Opportunity Employer, believes that students learn best in a diverse, inclusive community and is therefore committed to academic excellence through diversity in its faculty, staff, and students. Thus, we seek candidates who are committed to Bucknell's efforts to create a climate that fosters the growth and development of a diverse student body. We welcome applications from members of groups that have been historically underrepresented in higher education.

Cal Poly, San Luis Obispo
Electrical Engineering
Tenure-Track Faculty

COMPUTER ENGINEERING - The Electrical Engineering Depart & Computer Engineering Prog at Cal Poly, San Luis Obispo, invite applications for a full-time, tenure-track Computer Engineering faculty position at the **Assistant Professor** rank. The projected start date is September 15, 2014 or earlier. For details, qualifications, and application instructions (online application required), visit WWW.CALPOLYJOBS.ORG and apply to requisition #102950. Application review begins Jan. 6, 2014. EEO.

California State University, Fullerton
Department of Computer Science
Assistant Professor

The Department of Computer Science invites applications for **three tenure-track positions** at the **Assistant Professor** level starting fall 2014. For a complete description of the department, the position, desired specialization and other qualifications, please visit http://diversity.fullerton.edu/.

California State University, Sacramento

Department of Computer Science

Two Tenure-Track Assistant Professor positions

California State University, Sacramento, Department of Computer Science. Two Tenure-Track Assistant Professor positions to begin August 27, 2014 (open until filled). One position is in Information Assurance and Computer Security, and the other in Computer Games and Graphics. Ph.D. in Computer Science, Computer Engineering, or closely related field required by the time of appointment. For detailed position information, including application procedure, please see http://www.csus.edu/hr/faculty/vacancies.htm or http://www.ecs.csus.edu/csc. Screening will begin March 1, 2014, and continue until positions are filled. To apply, send cover letter, current vita including a list of publications, statement of research and teaching interests, transcripts of all college work including undergraduate work (unofficial copies acceptable until invited for interview), names and phone numbers of at least three recent references familiar with teaching and research potential to: Search Committee, Department of Computer Science, California State University, Sacramento, 6000 J Street, Sacramento, CA 95819-6021; or cscsearch@ecs.csus.edu. Incomplete applications will not be considered. AA/EEO employer. Clery Act statistics available. Mandated reporter requirements. Criminal background check may be required.

California State University, San Bernardino

School of Computer Science and Engineering

Assistant Professor

The **School of Computer Science and Engineering** invites applications for a tenure track position at the **Assistant Professor** level. Candidates must have a Ph.D. or an earned Doctorate in Computer Science or a closely related field. We are particularly interested in candidates with strengths in computer systems, software engineering, and computer security. Other areas of computer science will also be considered. The position is primarily to support the B.S. in Computer Science (ABET accredited) and B.A. in Computer Systems programs. In addition, the school offers the degrees B.S. in Computer Engineering, B.S. in Bioinformatics and M.S. in Computer Science.

The candidate must display potential for excellence in teaching and scholarly work. The candidate is expected to supervise student research at both the undergraduate and graduate levels, and to actively participate in other types of academic student advising. The candidate will actively contribute to the School's curriculum development. . The candidate will serve the School, College and University, as well as the community and the profession.

Women and underrepresented minorities are strongly encouraged to apply. For more information about the School of Computer Science and Engineering, please visit http://cse.csusb.edu

SALARY: Dependent on qualifications and experience.

BENEFITS: Generous medical, dental, and vision benefits and support for moving expenses available.

DEADLINE AND APPLICATION PROCESS: Applicants should submit a curriculum vitae, statement of teaching philosophy, description of research interest, an official copy of most recent transcripts, contact information for three references, and have three letters of recommendation sent separately. Review of applications will begin January 15, 2014, and will continue until the position is filled. The position will start in September 2014.

Please send all materials to:

Dr. Kerstin Voigt, Director
School of Computer Science and Engineering
California State University San Bernardino
5500 University Parkway
San Bernardino, CA 92407
Email Address: kvoigt@csusb.edu

Dartmouth College

Department of Computer Science

Assistant Professor of Computer Science: Computer Graphics/Digital Arts

The **Dartmouth College Department of Computer Science** invites applications for a tenure-track faculty position at the level of assistant professor. We seek candidates who will be excellent researchers and teachers in the areas of computer graphics and/or digital arts, although outstanding candidates in any area will be considered. We particu-

FIU FLORIDA INTERNATIONAL UNIVERSITY

Florida International University is a comprehensive university offering 340 majors in 188 degree programs in 23 colleges and schools, with innovative bachelor's, master's and doctoral programs across all disciplines including medicine, public health, law, journalism, hospitality, and architecture. FIU is Carnegie-designated as both a research university with high research activity and a community-engaged university. Located in the heart of the dynamic south Florida urban region, our multiple campuses serve over 50,000 students, placing FIU among the ten largest universities in the nation. Our annual research expenditures in excess of $100 million and our deep commitment to engagement have made FIU the go-to solutions center for issues ranging from local to global. FIU leads the nation in granting bachelor's degrees, including in the STEM fields, to minority students and is first in awarding STEM master's degrees to Hispanics. Our students, faculty, and staff reflect Miami's diverse population, earning FIU the designation of Hispanic-Serving Institution. At FIU, we are proud to be 'Worlds Ahead'! For more information about FIU, visit ***fiu.edu***.

The School of Computing and Information Sciences at Florida International University seeks candidates for tenure-track and tenured faculty positions at all levels.

Open-Rank Tenure Track/Tenured Positions (Job ID# 506754)

We seek outstanding candidates in all areas of Computer Science and researchers in the areas of compilers and programming languages, computer architecture, databases, information retrieval and big data, natural language processing, and health informatics, are particularly encouraged to apply. Candidates from minority groups are encouraged to apply. Preference will be given to candidates who will enhance or complement our existing research strengths.

Ideal candidates for junior positions should have a record of exceptional research in their early careers. Candidates for senior positions must have an active and proven record of excellence in funded research, publications, and professional service, as well as a demonstrated ability to develop and lead collaborative research projects. In addition to developing or expanding a high-quality research program, all successful applicants must be committed to excellence in teaching at both the graduate and undergraduate levels. An earned Ph.D. in Computer Science or related disciplines is required.

Florida International University (FIU) is the state university of Florida in Miami. It is ranked by the Carnegie Foundation as a comprehensive, doctoral research university with high research activity. The School of Computing and Information Sciences (SCIS) is a rapidly growing program of excellence at the University, with 36 faculty members and over 1,800 students, including 80 Ph.D. students. SCIS offers B.S., M.S., and Ph.D. degrees in Computer Science, an M.S. degree in Telecommunications and Networking, and B.S., B.A., and M.S. degrees in Information Technology. SCIS has received approximately $19.6M in the last four years in external research funding, has 14 research centers/clusters with first-class computing infrastructure and support, and enjoys broad and dynamic industry and international partnerships.

HOW TO APPLY: Applications, including a letter of interest, contact information, curriculum vitae, academic transcript, and the names of at least three references, should be submitted directly to the **FIU Careers Website at *careers.fiu.edu***; refer to Job ID# 506754. The application review process will begin on January 1st, 2014, and will continue until the position is filled. Further information can be obtained from the School website ***http://www.cis.fiu.edu***, or by e-mail to ***recruit@cis.fiu.edu***.

FIU is a member of the State University System of Florida and is an Equal Opportunity, Equal Access Affirmative Action Employer.

larly seek candidates who will be integral members of the Digital Arts program and help lead, initiate, and participate in collaborative research projects both within Computer Science and involving other Dartmouth researchers, including those in other Arts & Sciences departments, Dartmouth's Geisel School of Medicine, and Thayer School of Engineering.

The department is home to 17 tenured and tenure-track faculty members and two research faculty members. Research areas of the department encompass the areas of systems, security, vision, digital arts, algorithms, theory, robotics, and computational biology. The Computer Science department is in the School of Arts & Sciences, and it has strong Ph.D. and M.S. programs and outstanding undergraduate majors. Digital Arts at Dartmouth is an interdisciplinary program housed in the Computer Science department, working with several other departments, including Studio Art, Theater, and Film and Media Studies. The department is affiliated with Dartmouth's M.D.-Ph.D. program and has strong collaborations with Dartmouth's other schools.

Dartmouth College, a member of the Ivy League, is located in Hanover, New Hampshire (on the Vermont border). Dartmouth has a beautiful, historic campus, located in a scenic area on the Connecticut River. Recreational opportunities abound in all four seasons.

With an even distribution of male and female students and over one third of the undergraduate student population members of minority groups, Dartmouth is committed to diversity and encourages applications from women and minorities.

To create an atmosphere supportive of research, Dartmouth offers new faculty members grants for research-related expenses, a quarter of sabbatical leave for each three academic years in residence, and flexible scheduling of teaching responsibilities.

Applicants are invited to submit application materials via Interfolio at http://apply.interfolio.com/23489. Upload a CV, research statement, and teaching statement, and request at least four references to upload letters of recommendation, at least one of which should comment on teaching. Email facsearch14@cs.dartmouth.edu with any questions.

Application review will begin November 1, 2013, and continue until the position is filled.

Dartmouth College
Department of Computer Science
Assistant Professor of Computer Science: Machine Learning

The **Dartmouth College Department of Computer Science** invites applications for a tenure-track faculty position at the level of assistant professor. We seek candidates who will be excellent researchers and teachers in the area of machine learning, although outstanding candidates in any area will be considered. We particularly seek candidates who will help lead, initiate, and participate in collaborative research projects both within Computer Science and involving other Dartmouth researchers, including those in other Arts & Sciences departments, Dartmouth's Geisel School of Medicine, Thayer School of Engineering, and Tuck School of Business.

The department is home to 17 tenured and tenure-track faculty members and two research

---

UNIVERSITY of ROCHESTER

Tenure-Track Faculty Positions in Interdisciplinary Research in Data Science

The University of Rochester has made data science the centerpiece of its 5-year strategic plan, committing to 20 new faculty lines in diverse areas, a new building, and the establishment of the Institute for Data Science. We are currently seeking applicants for tenure track positions in interdisciplinary research areas within data science. This search complements department-specific searches in data science currently underway.

The interdisciplinary search focuses on recruiting candidates who are excited about engaging in collaborative research that connects advances in computational models and methods to other fields of engineering or life, social, or physical sciences. Successful candidates will receive a primary appointment in one of the departments supporting the search, and a secondary appointment in at least one other department. Departments supporting this search are Biomedical Engineering, Biology, Biostatistics & Computational Biology, Brain & Cognitive Sciences, Chemistry, Computer and Electrical Engineering, Computer Science, Earth & Environmental Sciences, Economics, Linguistics, Mathematics, Physics & Astronomy, and Political Science.

Focus areas for this year's interdisciplinary search are:

- *Fundamental Methods in Machine Learning, Network Science, and Statistics*
  Research on general computational methods for constructing systems for classification, prediction, classification, clustering, and related tasks from large-scale data. We are particularly interested in work on analyzing complex relational data using network models.
- *Computational Linguistics, Computer Vision, and Computational Models of Human Perception*
  Research in computational linguistics or computer vision, with a particular interest in work that spans shallow and deep semantic processing, and/or relates computational models to physiological models of perception. Candidates should have extensive experience with natural corpora and other rich databases.
- *Computational Biology and Computational Bioengineering*
  Research on computational approaches to analyze large, complex data sets in biology and biomedical engineering. Potential areas of research may include functional and evolutionary genomics, proteomics and protein folding, systems biology, multi-scale modeling in bioengineering, or multimodal bio-imaging informatics.
- *Global Biogeochemistry*
  Research that integrates biotic (e.g. microbial), chemical, and geological processes for an interdisciplinary research focus on understanding global geochemical cycling processes and/or global climate change. Applicants should have a strong computational and/or modeling component to their research aimed at mining, integrating, and/or interpreting large data sets.

Instructions for applying for this search appear at:

http://www.rochester.edu/rocdata/recruit/interdiscplinary

Applicants should hold a PhD and will be required to supply a set of refereed scholarly publications, names of references, and research and teaching statements. The application will ask applicants to select a set of disciplines most relevant to their research area. Review of applications at any rank will begin immediately and continue until the positions are filled. For full consideration, applications should be received by January 1st, 2014.

The University of Rochester is a private, Tier I research institution located in western New York State. It consistently ranks among the top 30 institutions, both public and private, in federal funding for research and development. The university has made substantial investments in computing infrastructure through the Center for Integrated Research Computing (CIRC) and the Health Sciences Center for Computational Innovation (HSCCI). The university includes the Eastman School of Music and the University of Rochester Medical Center, a major medical school, research center, and hospital system. The greater Rochester area is home to over a million people, including 80,000 students who attend the 8 colleges and universities in the region.

The University of Rochester has a strong commitment to diversity and actively encourages applications from candidates from groups underrepresented in higher education. The University is an Equal Opportunity Employer.

faculty members. Research areas of the department encompass the areas of systems, security, vision, digital arts, algorithms, theory, robotics, and computational biology. The Computer Science department is in the School of Arts & Sciences, and it has strong Ph.D. and M.S. programs and outstanding undergraduate majors. The department is affiliated with Dartmouth's M.D.-Ph.D. program and has strong collaborations with Dartmouth's other schools.

Dartmouth College, a member of the Ivy League, is located in Hanover, New Hampshire (on the Vermont border). Dartmouth has a beautiful, historic campus, located in a scenic area on the Connecticut River. Recreational opportunities abound in all four seasons.

With an even distribution of male and female students and over one third of the undergraduate student population members of minority groups, Dartmouth is committed to diversity and encourages applications from women and minorities.

To create an atmosphere supportive of research, Dartmouth offers new faculty members grants for research-related expenses, a quarter of sabbatical leave for each three academic years in residence, and flexible scheduling of teaching responsibilities.

Applicants are invited to submit application materials via Interfolio at http://apply.interfolio.com/23502. Upload a CV, research statement, and teaching statement, and request at least four references to upload letters of recommendation, at least one of which should comment on teaching. Email facsearch14@cs.dartmouth.edu with any questions.

Application review will begin November 1, 2013, and continue until the position is filled.

Dartmouth College

Department of Computer Science

Assistant Professor of Computer Science: Theory/Algorithms

The **Dartmouth College Department of Computer Science** invites applications for a tenure-track faculty position at the level of assistant professor. We seek candidates who will be excellent researchers and teachers in the area of theoretical computer science, including algorithms, although outstanding candidates in any area will be considered. We particularly seek candidates who will help lead, initiate, and participate in collaborative research projects both within Computer Science and involving other Dartmouth researchers, including those in other Arts & Sciences departments, Dartmouth's Geisel School of Medicine, Thayer School of Engineering, and Tuck School of Business.

The department is home to 17 tenured and tenure-track faculty members and two research faculty members. Research areas of the department encompass the areas of systems, security, vision, digital arts, algorithms, theory, robotics, and computational biology. The Computer Science department is in the School of Arts & Sciences, and it has strong Ph.D. and M.S. programs and outstanding undergraduate majors. The department is affiliated with Dartmouth's M.D.-Ph.D. program and has strong collaborations with Dartmouth's other schools.

Dartmouth College, a member of the Ivy League, is located in Hanover, New Hampshire (on the Vermont border). Dartmouth has a beautiful, historic campus, located in a scenic area on the Connecticut River. Recreational opportunities abound in all four seasons.

With an even distribution of male and female students and over one third of the undergraduate student population members of minority groups, Dartmouth is committed to diversity and encourages applications from women and minorities.

To create an atmosphere supportive of research, Dartmouth offers new faculty members grants for research-related expenses, a quarter of sabbatical leave for each three academic years in residence, and flexible scheduling of teaching responsibilities.

Applicants are invited to submit application materials via Interfolio at http://apply.interfolio.com/23503. Upload a CV, research statement, and teaching statement, and request at least four references to upload letters of recommendation, at least one of which should comment on teaching. Email facsearch14@cs.dartmouth.edu with any questions.

Application review will begin November 1, 2013, and continue until the position is filled.

Drexel University

College of Computing & Informatics

Faculty Positions

Drexel University's new College of Computing & Informatics (cci.drexel.edu) invites applications for

multiple tenure-track faculty positions at all levels. The College offers graduate & undergraduate degrees in computer science, cybersecurity, informatics, info systems, info technology, library & info science, and software engineering. We seek candidates who can contribute to university-wide objectives in Energy, Health Sciences & Systems, Sustainability, Entrepreneurship, and Information & Society and align with strategic plans for the College and University (http://www.drexel.edu/strategicPlan/).

Areas of interest this year include (1) Security & Privacy (e.g., cryptography, cyber-policy & ethics); (2) Software & Systems Engineering (e.g., cloud & mobile computing, software quality, software process, architecture, & system administration); (3) Intelligent Systems (e.g., computer vision, machine learning, gaming, GIS); (4) Human-Centered Computing (e.g., HCI, socio-technical studies, eLearning, decision support, neuro/cognitive modeling); (5) Informatics & Data Science (e.g., eScience, databases, data mining, analytics & visualization); (6) Library & Information Science (e.g., archives, library systems, digital libraries, info policy, info behavior, & retrieval). Exceptional candidates in other areas will be considered.

Drexel is a private university committed to research with real-world applications. The university has over 25,000 students in 14 colleges and schools and offers about 200 degree programs. The College of Computing and Informatics has about 75 faculty and 2,300 students. Drexel has one of the largest and best known cooperative education programs with over 1,200 co-op employers. Drexel is located on Philadelphia's "Avenue of Technology" in the University City District and at the hub of the academic, cultural, and historical resources of the nation's sixth largest metropolitan region.

Review of applications begins immediately. To assure consideration, materials from applicants should be received by February 28, 2014. Successful applicants must demonstrate potential for research & teaching excellence in the environment of a major research university. To be considered, apply at https://www.drexeljobs.com, Requisition #5673.

Your application should consist of a cover letter, CV, & brief statements describing your research program & teaching interests. Letters of reference will be requested from the candidates who are invited for a campus interview. Electronic submissions in PDF format are required.

Drexel University is an Equal Opportunity/Affirmative Action Employer. The College of Computing & Informatics is especially interested in qualified candidates who can contribute to the diversity and excellence of the academic community. Background investigations are required for all new hires as a condition of employment, after the job offer is made. Employment will be contingent upon the University's acceptance of the results of the background investigation.

George Mason University
Volgenau School of Engineering
Tenure-Track Assistant Professor, Cyber Security Engineering

The George Mason University, Volgenau School of Engineering invites applications for a Tenure-Track position at the rank of Assistant Professor in the area of Cyber Security Engineering starting fall 2014. Cyber security engineering refers to the proactive engineering design of physical systems with cyber security incorporated.

Qualifications:
Minimum qualifications for the position include a Ph.D. in any engineering discipline with experience in cyber security; demonstrated potential for excellence in research; and a commitment to high-quality teaching. Successful applicants will need to demonstrate a planned research agenda in areas related to cyber security engineering and a commitment to obtain external funding.

The Volgenau School of Engineering has ongoing programs at the undergraduate and graduate levels in the areas of information security and assurance. These programs and related research are conducted within the departments of Applied Information Technology, Computer Science, and Electrical and Computer Engineering; and at the Center for Secure Information Systems. The Volgenau School is in the process of introducing a new undergraduate degree in Cyber Security Engineering. This program is an interdisciplinary degree that will reside at the school-level.

George Mason University is an innovative, entrepreneurial institution with national distinction in a range of academic fields, and has been ranked the number one "up-and-coming" university by the U.S. News and World Report. In 2013, the school is ranked sixth among U.S. schools for return on investment (ROI) by Payscale.com and Affordablecollegesonline.org/. Enrollment at Ma-

---

son is approximately 34,000, with students studying in over 198 degree programs. The Volgenau School of Engineering maintains close ties with the engineering community in northern Virginia and the metropolitan Washington, D.C., area in both industry and government. For more information about the Volgenau School of Engineering, please see http://volgenau.gmu.edu/.

For full consideration, please submit an online faculty application at http://jobs.gmu.edu for position number F9766z; and attach a curriculum vita, letter of intent, a statement of teaching and research, and the contact information for three professional references. Questions about the position should be directed to Professor Sushil Jajodia at jajodia@gmu.edu. Review of applications will commence January 31, 2014, and continue until the position is filled.

George Mason University is an affirmative action/equal opportunity employer encouraging diversity.

Hendrix College

Assistant Professor of Computer Science

Hendrix College invites applications to join our Computer Science program in August 2014 as a tenure-track Assistant Professor. The successful candidate will be committed to excellent teaching in a liberal arts environment and will sustain a research program involving undergraduates. For fullest consideration, submit an online application (https://academicjobsonline.org/ajo/jobs/3347) by December 1, 2013.

Iowa State University

Assistant Professor BIGDATA

The Department of Computer Science at Iowa State University seeks outstanding applicants for an Assistant Professor at the intersection of big data analytics, bioinformatics, and computational biology. Ph.D. in computer science, bioinformatics, computational biology, or a closely related field. ISU is a member of the prestigious Association of American Universities and is located in Ames, Iowa. http://www.iastate.edu/about/. For information on this vacancy or to apply online, visit: https://www.iastatejobs.com/ Vacancy #131181. ISU is an EO/AA employer.

Iowa State University

Tenure-Track or Tenured Assistant, Associate or Full Professor - BIGDATA MSR

Iowa State University has launched the Presidential High Impact Hires Initiative. As part of this initiative, the Computer Science Department and College of Liberal Arts and Sciences invites applications for a tenure-track or tenured position at Assistant, Associate or Full Professor level.

- ▸ Ph.D. in computer science, computer engineering, or a closely related field.
- ▸ Demonstrated track record in mining software repository (MSR) techniques for large-scale repositories.
- ▸ Demonstrated potential for outstanding research and publication.
- ▸ To be considered for Associate or Full Professor, there must be an outstanding research record as demonstrated by publications in peer-reviewed journals and at peer-reviewed conferences.
- ▸ To be considered for Full Professor, there must be an outstanding record in obtaining grants and contracts particularly those from prestigious national agencies.

ISU is a member of the prestigious Association of American Universities and is located in Ames, Iowa. If interested, visit: https://www.iastatejobs.com/. Look for Vacancy #131239. Information about ISU: http://www.iastate.edu/about/. ISU is an EO/AA employer.

Lamar University

Department of Computer Science
Assistant Professor

Lamar University's Department of Computer Science seeks applications for a tenure-track **Assistant Professor** position beginning fall 2014. Applicants must have a PhD in Computer Science and a strong commitment to teaching and research. Particular areas of interest include computer graphics and vision, high performance computing, computational science and scientific computing, security and privacy, data mining and cloud computing, and bioinformatics and computational biology. Lamar is an AA/EEO state supported university of approximately 14,500 students. It offers the B.S. and M.S. in Computer Science. There are 9 full-time faculty and 260 undergraduate and graduate majors.

Apply at http://jobs.lamar.edu/postings/1499 Review of applications will begin on March 1, 2014, and continue until the position is filled. If you have additional questions, please address them at dept_cs@lamar.edu

Max Planck Institute for Informatics

Junior Research Group Leaders in the Max Planck Center for Visual Computing and Communication

The Max Planck Institute for Informatics, as the coordinator of the Max Planck Center for Visual Computing and Communication (MPC-VCC), invites applications for **Junior Research Group Leaders in the Max Planck Center for Visual Computing and Communication**

The Max Planck Center for Visual Computing and Communications offers young scientists in information technology the opportunity to develop their own research program addressing important problems in areas such as

- ▸ image communication
- ▸ computer graphics
- ▸ geometric computing
- ▸ imaging systems
- ▸ computer vision
- ▸ human machine interface
- ▸ distributed multimedia architectures
- ▸ multimedia networking
- ▸ visual media security.

The center includes an outstanding group of faculty members at Stanford's Computer Science and El–ectrical Engineering Departments, the Max Planck Institute for Informatics, and Saarland University.

The program begins with a preparatory 1-2 year postdoc phase (**Phase P**) at the Max Planck Institute for Informatics, followed by a two-year appointment at Stanford University (**Phase I**) as a visiting assistant professor, and then a position at the Max Planck Institute for Informatics as a junior research group leader (**Phase II**). However, the program can be entered flexibly at each phase, commensurate with the experience of the applicant.

Applicants to the program must have completed an outstanding PhD. Exact duration of the preparatory postdoc phase is flexible, but we typically expect this to be about 1-2 years. Applicants who completed their PhD in Germany may enter Phase I of the program directly. Applicants for Phase II are expected to have completed a postdoc stay abroad and must have demonstrated their outstanding research potential and ability to successfully lead a research group.

Reviewing of applications will commence on **01 Jan 2014**. The final deadline is **31 Jan 2014**. Applicants should submit their CV, copies of their school and university reports, list of publications, reprints of five selected publications, names of 3-5 references, a brief description of their previous research and a detailed description of the proposed research project (including possible opportunities for collaboration with existing research groups at Saarbrücken and Stanford) to:

Prof. Dr. Hans-Peter Seidel
Max Planck Institute for Informatics,
Campus E 1 4, 66123 Saarbrücken, Germany
Email: mpc-vcc@mpi-inf.mpg.de

The Max Planck Center is an equal opportunity employer and women are encouraged to apply.

Additional information is available on the website http://www.mpc-vcc.de

Max Planck Institute for Software Systems

Tenure-track / Tenured Positions

Applications are invited for tenure-track and tenured faculty positions in all areas related to the study, design, and engineering of software systems. These areas include, but are not limited to, security and privacy, embedded and mobile systems, social computing, large-scale data management and machine learning, programming languages and systems, software verification and analysis, parallel and distributed systems, storage systems, and networking.

A doctoral degree in computer science or related areas and an outstanding research record are required. Successful candidates are expected to build a team and pursue a highly visible research agenda, both independently and in collaboration with other groups. Senior candidates must have demonstrated leadership abilities and recognized international stature.

MPI-SWS, founded in 2005, is part of a network of eighty Max Planck Institutes, Germany's premier basic research facilities. MPIs have an established record of world-class, foundational research in the fields of medicine, biology, chemistry, physics, technology and humanities. Since 1948, MPI researchers have won 17 Nobel prizes. MPI-SWS aspires to meet the highest standards of excellence and international recognition with its research in software systems.

To this end, the institute offers a unique environment that combines the best aspects of a university department and a research laboratory:

a) Faculty receive generous base funding to build and lead a team of graduate students and post-docs. They have full academic freedom and publish their research results freely.

b) Faculty supervise doctoral theses, and have the opportunity to teach graduate and undergraduate courses.

c) Faculty are provided with outstanding technical and administrative support facilities as well as internationally competitive compensation packages.

MPI-SWS currently has 10 tenured and tenure-track faculty and 40 doctoral and post-doctoral researchers. The institute is funded to support 17 faculty and up to 100 doctoral and post-doctoral positions. Additional growth through outside funding is possible. We maintain an open, international and diverse work environment and seek applications from outstanding researchers regardless of national origin or citizenship. The working language is English; knowledge of the German language is not required for a successful career at the institute.

The institute is located in Kaiserslautern and Saarbruecken, in the tri-border area of Germany, France and Luxembourg. The area offers a high standard of living, beautiful surroundings and easy access to major metropolitan areas in the center of Europe, as well as a stimulating, competitive and collaborative work environment. In immediate proximity are the MPI for Informatics, Saarland University, the Technical University of Kaiserslautern, the German Center for Artificial Intelligence (DFKI), and the Fraunhofer Institutes for Experimental Software Engineering and for Industrial Mathematics.

Qualified candidates should apply online at https://apply.mpi-sws.org. The review of applications will begin on December 15, 2013, and applicants are strongly encouraged to apply by that date; however, applications will continue to be accepted through January 2014.

The institute is committed to increasing the representation of minorities, women and individuals with physical disabilities in Computer Science. We particularly encourage such individuals to apply. You can find more information about the positions at: http://www.mpi-sws.org/index.php?n=careers/tenure-track

McMaster University

Department of Computing and Software

Tenure-Track Faculty Position

Ranked among the top engineering schools in Canada and worldwide, the Faculty of Engineering plays a key role in helping McMaster University earn its well-deserved reputation as one of Canada's most innovative universities in learning and research.

The McMaster Faculty of Engineering has a reputation for innovative programs, cutting-edge research, leading faculty, and aspiring students. It has earned a strong reputation as a centre for academic excellence and innovation. The Faculty has approximately 150 faculty members, along with close to 4,000 undergraduate and 750 graduate students.

The Department of Computing and Software at McMaster University seeks outstanding candidates for a tenure-track faculty position. The appointment is intended to be at the Assistant or Associate Professor level. Applicants with a doctorate in Computer Science or Software Engineering (or a related area) at the time of appointment are encouraged to apply. The Department invites applications from exceptional candidates in all areas, including those with expertise in: big data, cyber-physical systems, digital media and human-computer interaction, high performance computing, mobile computing, optimization, and systems.

The potential to develop a strong research program and become an excellent teacher is crucial. Successful candidates will be expected to attract external research funding, pursue industrial collaboration if appropriate, actively recruit and supervise graduate students, and teach at both the undergraduate and graduate levels. Registration or eligibility for registration by the Professional Engineers of Ontario will be considered an asset.

Salary and rank are commensurate with experience and qualifications. Applications, including a CV, a statement detailing teaching and research interests, and the names of at least three referees should be sent electronically to Laurie LeBlanc at leblanl@mcmaster.ca or in hard copy to:

Chair
Department of Computing and Software,
ITB 202
McMaster University
1280 Main Street West
Hamilton, ON L8S 4K1

Applications review will begin immediately and the appointment will ideally commence July 1, 2014. However, applications will be accepted until the position is filled.

Note: All qualified candidates are encouraged to apply. However, Canadian citizens and permanent residents will be considered first for these positions. McMaster University is strongly committed to employment equity within its community, and to recruiting a diverse faculty and staff. The University encourages applications from all qualified candidates, including women, members of visible minorities, Aboriginal peoples, members of sexual minorities and persons with disabilities.

New York University

Courant Institute of Mathematical Sciences
Arts and Science

Clinical Assistant/Associate Professor Position in Computer Science

The Computer Science Department at New York University has an opening for a Clinical Assistant or Associate Professor position to start September 1, 2014, subject to budgetary and administrative approval. This is a full-time non-tenured, non-tenure-track three-year contract faculty position which is potentially renewable. The main duty is to teach three courses during each of the fall and spring semesters in the department's undergraduate program and additionally to participate in curricular development, program administration, and other educational activities. Applicants should have an M.S. or Ph.D. in Computer Science or a related field. To apply, please arrange for a CV and for three letters of recommendation to be sent by email to jobs@cs.nyu.edu. To guarantee full consideration, complete applications should be received by March 15, 2014. However, all candidates will be considered to the extent feasible until the position is filled. NYU is an Equal Opportunity/Affirmative Action Employer.

Northwestern University

Department of Electrical Engineering and Computer Science

Multiple Openings in Theoretical Computer Science

Start Date: Fall 2014

The Department of Electrical Engineering and Computer Science at Northwestern University invites applications from exceptionally qualified candidates for multiple faculty positions in theoretical computer science to start fall 2014. The positions are open for all professorial ranks and all areas of theoretical computer science.

The 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 package are negotiable and will be competitive.

Northwestern EECS consists of over 50 faculty members of international prominence whose interests span a wide range. Northwestern University is located in Evanston, Illinois on the shores of Lake Michigan just north of Chicago. Further information about the Department and the University is available at http://www.eecs.northwestern.edu and http://www.northwestern.edu.

To ensure full consideration, applications should be received by December 1, 2013. Applications will be accepted until the positions are filled.

To apply, please visit http://eecs.northwestern.edu/academic-openings.html for full instructions on uploading. Applicants are asked to submit (1) a cover letter indicating the rank applied for, (2) the names of between five and eight references for the rank of Full or Associate Professor or at least three references for the rank of Assistant Professor, and (3) a curriculum vitae. The search committee will request letters from the references and may request (4) statements of research and teaching interests and (5) three representative publications. For assistance with application materials or general questions, contact tcs.facultysearch.2013@eecs.northwestern.edu.

Northwestern University is an equal opportunity, affirmative action employer. Qualified women and minorities are encouraged to apply. It is the policy of Northwestern University not to discriminate against any individual on the basis of race, color, religion, national origin, gender, sexual orientation, marital status, age, disability, citizenship, veteran status, or other protected group status. Hiring is contingent upon eligibility to work in the United States.

Princeton University

Computer Science Department

Part-Time or Full-Time Lecturer

The Department of Computer Science seeks applications from outstanding teachers to assist the faculty in teaching our introductory course sequence or some of our upper-level courses.

Depending on the qualifications and interests of the applicant, job responsibilities will include such activities as teaching recitation sections and supervising graduate-student teaching assistants; grading problem sets and programming assignments; supervising students in the grading of problem sets and programming assignments; developing and maintaining online curricular material, classroom demonstrations, and laboratory exercises; and supervising undergraduate research projects. An advanced degree in computer science, or related field, is required (PhD preferred).

The position is renewable for 1-year terms, up to six years, depending upon departmental need and satisfactory performance.

To apply, please submit a cover letter, CV, and contact information for three references to (https://jobs.cs.princeton.edu/lecturer)

Princeton University is an equal opportunity employer and complies with applicable EEO and affirmative action regulations.

Princeton University

Computer Science

Postdoctoral Research Associate

The Department of Computer Science at Princeton University is seeking applications for postdoctoral or more senior research positions in theoretical computer science. Positions are for one year with the possibility of renewal.

Candidates should have a PhD in Computer Science or a related field by August 2014. To ensure full consideration, we encourage candidates to complete their applications, (including letters of recommendation) by December 10, 2013. Applicants should submit a CV and research statement, and contact information for three references. Princeton University is an equal opportunity employer and complies with applicable EEO and affirmative action regulations. Apply to: http://jobs.princeton.edu/. Req.# 1300791

Purdue University School of ECE

Faculty Opening in Computer Engineering

As part of the Engineering Strategic Growth Initiative, Purdue University is seeking to fill a faculty position in computer engineering within the School of Electrical and Computer Engineering. The Computer Engineering area of the school (https://engineering.purdue.edu/ECE/Research/Areas/CompEngr) has twenty faculty members with active research programs in areas such as AI, architecture, compilers, computer vision, distributed systems, embedded systems, graphics, haptics, HCI, machine learning, multimedia systems, networking, networking applications, NLP, OS, robotics, software engineering, and visualization. The new hire will join a strong group of computer engineering faculty and will help shape Purdue's vision and research/education agenda.

Candidates for tenure-track position at the Assistant Professor level will be considered. Outstanding candidates in all areas of computer science and engineering including, but not limited to, cloud computing, distributed systems, embedded systems, intelligent systems, mobile computing, and security will be considered.

Eligible candidates are required to have a PhD degree in computer science, computer engineering, or a closely-related discipline, have demonstrated potential for excellence in research, and be committed to excellence in teaching.

The successful candidate will have a distinguished academic record, will develop a strong, independent research program, be expected to teach undergraduate and/or graduate courses, and advise students.

Salary and benefits are highly competitive. Submit applications online at

https://engineering.purdue.edu/Engr/AboutUs/Employment/Applications.

The application should include a cover letter, a complete and detailed vitae, and statements of research and teaching interests. Also, include names, addresses, telephone numbers, and email addresses for three or more references.

Review of applications will begin on December 1, 2013, and will continue until filled. Inquiries may be sent to ece-cearea-search@ecn.purdue.edu. Applications will be considered as they are received, but, for full consideration, should arrive by January 15, 2014.

A background check will be required for employment in this position. Purdue University is an Equal Opportunity/Equal Access/Affirmative Action employer fully committed to achieving a diverse workforce.

Purdue University School of ECE

Faculty Position in Human-Centered Computing

The School of Electrical and Computer Engineering at Purdue University invites applications for two faculty positions at the assistant, associate or full rank in human-centered computing, including but not limited to visualization, visual analytics, human-computer interaction (HCI), imaging, and graphics. The Computer Engineering area of the school (https://engineering.purdue.edu/ECE/Research/Areas/CompEngr) has twenty faculty members with active research programs in areas such as AI, architecture, compilers, computer vision, distributed systems, embedded systems, graphics, haptics, HCI, machine learning, multimedia systems, networking, networking applications, NLP, OS, robotics, software engineering, and visualization. The Communications, Networks and Signal Processing area of the school (https://engineering.purdue.edu/ECE/Research/Areas/CommSigP) has eighteen faculty members with active research programs in areas such as wireless mobile and PCS communication, smart antennas, GPS, radar, speech recognition and synthesis, image processing and pattern recognition, image quality, image rendering, document imaging, image analysis, remote sensing, local and wide-area computer networks, and multimedia communication, security, and processing.

Eligible candidates are required to have a Ph.D. or equivalent doctoral level degree in computer science/engineering or a closely related field and a significant demonstrated research record commensurate with the level of the position for which they are applying.

The successful candidate will have a distinguished academic record, will develop a strong, independent research programs, will teach undergraduate and/or graduate level courses, and will advise students. Applications should consist of a cover letter, a CV, research and teaching statements, names and contact information for at least three references, and URLs for three to five online papers. Applications should be submitted to https://engineering.purdue.edu/Engr/AboutUs/Employment/Applications.

Review of applications will begin on December 1, 2013. Inquiries may be sent to ece-hccsearch@ecn.purdue.edu. Applications will be considered as they are received, but for full consideration should arrive by January 15, 2014. A background check will be required for employment in this position. Purdue University is an equal opportunity, equal access, affirmative action employer fully committed to achieving a diverse workforce.

Qatar University

Associate/Full Research Professor

Research Faculty Positions at Qatar University

Qatar University invites applications for research faculty positions at the level of associate of full professor to begin on September 2014. Candidates in the following fields will be considered:

- Cyber security
- Bioinformatics

Candidates will cultivate and lead large-scale research projects at the KINDI Lab for Computing Research in the areas of cloud computing security, privacy and cancer informatics.

Qatar University offers a competitive benefits package including a 3-year renewable contract, tax free salary, free furnished accommodation, and more.

Apply by posting you application before Feb-

ruary 28, 2014 on the QU online recruitment system http://careers.qu.edu.qa under "College of Engineering".

Queens College
Tenure-Track Assistant Professor - Computer Science

The Department of Computer Science at Queens College of CUNY is accepting applications for a tenure-track position in high performance computing at the Assistant Professor level starting Fall 2014. Consult http://www.cs.qc.cuny.edu for further information.

State University of New York at Binghamton
Department of Computer Science
Four Tenure-Track Assistant Professor Positions

Applications are invited for four tenure-track Assistant Professor positions beginning Fall 2014 with specializations in: (a) cybersecurity (three positions) and, (b) embedded systems programming/design with an emphasis on energy optimization (one position). The Department has established graduate and undergraduate programs, including 60 full-time PhD students. Junior faculty have a significantly reduced teaching load for at least the first three years. Please indicate your teaching and research areas of interest in a single sentence on your cover letter.

Further details and application information are available at:

http://www.binghamton.edu/cs

Applications will be reviewed until positions are filled. First consideration will be given to applications received by **February 17, 2014**.

We are an EE/AA employer.

The Catholic University of America
Tenure Track Assistant Professor in Computer Science

The Department of Electrical Engineering and Computer Science at the Catholic University of America invites applications for a tenure-track faculty position at the Assistant Professor Level, beginning August 2014. All areas in computer science will be given consideration, with a particular emphasis on network and information security. For full consideration, complete applications should be received by January 10, 2014. Additional information and application instructions can be found at http://engineering.cua.edu/facultyapp/eecs/

University of California, Riverside
Faculty Positions in the Department of Computer Science and Engineering

The Department of Computer Science and Engineering, University of California, Riverside invites applications for **two tenure-track and one tenured** faculty positions beginning the 2014/15 academic year. Candidates are sought in the following areas of research: (1) Networks, Operating/Distributed Systems, and Cyber-security; (2) High-Performance and Scientific Computing; and (3) Low Power Computer Design and Architecture. The first two searches are focused on the Assistant Professor level. The third search is focused on the senior level and will be interdisciplinary with the successful candidate affiliated with one or more engineering departments in the College. Exceptional candidates in all areas and at all levels will be considered. Positions require a Ph.D. in Computer Science (or in a closely related field) at the time of employment. Junior candidates must show outstanding research, teaching and graduate student mentorship potential. Exceptional senior candidates with outstanding research, teaching, and graduate student mentorship records will be considered. Salary level will be competitive and commensurate with qualifications and experience.

Details and application materials can be found at www.engr.ucr.edu/facultysearch. Full consideration will be given to applications received by January 1, 2014. We will continue to consider applications until the positions are filled. For inquiries and questions, please contact us at search@cs.ucr.edu. EEO/AA employer.

University of Colorado, Boulder
Assistant Professor

The Department of Computer Science (CS) at the University of Colorado Boulder seeks outstanding candidates, for a tenure-track position, with expertise in both machine learning and optimization. The opening is targeted at the level of Assistant Professor, although exceptional senior candidates at higher ranks may be considered.

We seek candidates whose primary research areas lie at the intersection of machine learning and numerical optimization, and whose research addresses challenges in theory, algorithms, implementation, and application of problems in optimization and machine learning. Candidates should demonstrate excellence in both research and teaching, have

a strong interest in interdisciplinary collaboration, and aim to lead a highly visible, externally funded research program.

Applications must be submitted online at http://www.jobsatcu.com/postings/73978

The University of Colorado is an Equal Opportunity/Affirmative Action employer.

University of Delaware
Department of Computer and Information Sciences
Tenure-track Assistant Professor in Big Data

Applications are invited for a tenure-track assistant professor position in Big Data (broadly defined) to begin Fall 2014. We seek innovative individuals, who have demonstrated excellence in research and drive to become leaders in their fields while engaging in high-quality teaching and mentoring. For information and application procedures, please visit www.udel.edu/udjobs.

The UNIVERSITY OF DELAWARE is an Equal Opportunity Employer and encourages applications from Minority Group Members and Women.

University of Illinois Springfield
Assistant Professor Computer Science

The Computer Science Department at the University of Illinois Springfield (UIS) invites applications for a beginning assistant professor, tenure track position to begin August, 2014. A Ph.D. in Computer Science or closely related field is required. The position involves graduate and undergraduate teaching, supervising student research, and continuing your research. Many of our classes are taught online. All areas of expertise will be considered, but the ability to teach core computer science is of special interest for the Department. Review of applications will begin on January 30, 2014 and continue until the position is filled or the search is terminated. Please send your vita and contact information for three references to Chair Computer Science Search Committee; One University Plaza; UHB 3100; Springfield, IL 62703-5407.

Located in the state capital, the University of Illinois Springfield is one of three campuses of the University of Illinois. The UIS campus serves approximately 5,000 students in 23 undergraduate and 21 graduate degree programs. The academic curriculum of the campus emphasizes a strong liberal arts core, an array of professional programs, extensive opportunities in experiential education, and a broad engagement in public affairs issues of the day. The campus offers many small classes, substantial student-faculty interaction, and a rapidly evolving technology enhanced learning environment. Its diverse student body includes traditional, non-traditional, and international students. Twenty-five percent of majors are in 17 undergraduate and graduate online degree programs and the campus has received several national awards for its implementation of online learning. UIS faculty are committed teachers, active scholars, and professionals in service to society. You are encouraged to visit the university web page at http://www.uis.edu and the department web page at http://csc.uis.edu . UIS is an affirmative action/equal opportunity employer with a strong institutional commitment to recruitment and retention of a diverse and inclusive campus community. Women, minorities, veterans, and persons with disabilities are encouraged to apply.

University of Maryland, Baltimore County
Computer Science and Electrical Engineering Department
Two Tenure Track Assistant Professor Positions, Computer Science

We invite applications for two tenure track positions in Computer Science at the rank of Assistant Professor to begin in August 2014. All areas will be considered, but we are especially interested in candidates in systems, security, or data analytics. Unusually strong candidates at the Associate Professor level will be considered. Submit a cover letter, brief statement of teaching and research experience and interests, CV, and three letters of recommendation. See http://csee.umbc.edu/about/jobs/ for more information about this search and concurrent searches for a tenure track position in Electrical and Computer Engineering and a Professor of the Practice position in Computer Science. UMBC is an AA/EOE.

University of Miami

College of Arts and Sciences

Computational Neuroscience Faculty Position

The College of Arts and Sciences at the **University of Miami** invites applications and nominations for one open-rank, **Assistant, Associate, or full Professor position in the Department of Computer Science** starting August 2014. Candidates must possess a Ph.D. in Computer Science or in a closely-related discipline with strong research expertise in areas related to Computational Neuroscience.

The College is strongly committed to Brain Science and has recently acquired a research-dedicated 3 Tesla fMRI for human brain imaging, which is available as a resource. Faculty in the Department of Psychology and other College departments and/or the School of Medicine are available for collaboration.

The successful candidate will be expected to teach at both undergraduate and graduate levels and to develop and maintain an internationally recognized research program. To be considered at Associate or Full Professor level the candidate must have a proven record of successful independent teaching and research.

Applicants should submit a cover letter, CV, research plan, statement of teaching philosophy, sample preprints or reprints, and the names of at least three references online to http://www.cs.miami.edu/search/. Review of applications will begin November 15, 2013 and continue until the position is filled. Information about the College can be found at http://www.as.miami.edu/.

The University of Miami offers competitive salaries and a comprehensive benefits package including medical and dental benefits, tuition remission, vacation, paid holidays and much more. The University of Miami is an equal opportunity/affirmative action employer that values diversity and has progressive work-life policies. Women, persons with disabilities, and members of other underrepresented groups are encouraged to apply.

University of Rochester

Department of Computer Science

Faculty Positions in Computer Science: Experimental Systems and Data Science

The **University of Rochester Department of Computer Science** seeks applicants for multiple tenure track positions in the broad areas of experimental systems and data science research (including but not exclusively focused on very large data-driven systems, machine learning and/or optimization, networks and distributed systems, operating systems, sustainable systems, security, and cloud computing). Candidates must have a PhD in computer science or a related discipline.

Apply online at
https://www.rochester.edu/fort/csc

Consideration of applications at any rank will begin immediately and continue until all interview slots are filled. Candidates should apply no later than January 1, 2014 for full consideration. Applications that arrive after this date incur a probability of being overlooked or arriving after the interview schedule is filled up.

The Department of Electrical and Computer Engineering (http://www.ece.rochester.edu/about/jobs.html) is also searching for a candidate broadly oriented toward data science. While the two searches are concurrent and plan to coordinate, candidates should apply to the department/s that best matches their academic background and interests.

The Department of Computer Science is a research-oriented department with a distinguished history of contributions in systems, theory, artificial intelligence, and HCI. We have a collaborative culture and strong ties to electrical and computer engineering, cognitive science, linguistics, and several departments in the medical center. Over the past decade, a third of the department's PhD graduates have won tenure-track faculty positions, and its alumni include leaders at major research laboratories such as Google, Microsoft, and IBM.

The University of Rochester is a private, Tier I research institution located in western New York State. It consistently ranks among the top 30 institutions, both public and private, in federal funding for research and development. The university has made substantial investments in computing infrastructure through the Center for Integrated Research Computing (CIRC) and the Health Sciences Center for Computational Innovation (HSCCI). Teaching loads are light and classes are small. Half of all undergraduates go on to post-graduate or professional education. The university includes the Eastman School of Music, a premiere music conservatory, and the University of Rochester Medical Center, a major medical school, research center, and hospital system. The greater Rochester area is home to over a million people, including 80,000 students who attend its 8 colleges and universities.

The University of Rochester has a strong commitment to diversity and actively encourages applications from candidates from groups underrepresented in higher education. The University is an Equal Opportunity Employer.

The University of Texas at San Antonio

Faculty Positions in Computer Science

The Department of Computer Science at The University of Texas at San Antonio invites applications for multiple tenure/tenure-track positions at all levels, starting Fall 2014. We are particularly interested in candidates in

- operating systems, distributed systems, or computer architecture at the assistant or associate professor level,
- big data/data science at the assistant or associate professor level, and
- cyber security (especially systems security) at the associate or full professor level.

Outstanding candidates in other areas will also be considered.

The Department of Computer Science currently has 22 faculty members and offers B.S., M.S., and Ph.D. degrees supporting a dynamic and growing program with 801 undergraduates and more than 190 graduate students, including 85 Ph.D. students. See http://www.cs.utsa.edu/fsearch for application instructions and additional information on the Department of Computer Science.

Screening of applications will begin on January 2, 2014 and will continue until the positions are filled or the search is closed. UTSA is an EO/AA Employer.

Chair of Faculty Search Committee
Department of Computer Science
The University of Texas at San Antonio
One UTSA Circle
San Antonio, TX 78249-0667
Phone: 210-458-4436

University of Wisconsin-Madison

Computer Sciences Department

Faculty Positions: Assistant Professors

The Computer Sciences Department at the University of Wisconsin-Madison has embarked on a multi-year effort to significantly enhance the strengths of the department. As part of the endeavor we have multiple openings for tenure-track Assistant Professors in any area of Computer Science.

Applicants must have a Ph.D. in Computer Science or in a closely related field prior to the start of the appointment. Successful candidates will show potential for developing an outstanding and highly visible scholarly research program, as well as excelling in undergraduate and graduate teaching.

Applicants should submit a curriculum vitae, a statement of research objectives and sample publications, and arrange to have at least three letters of reference sent directly to the department. Electronic submission of all application materials is preferred (see http://www.cs.wisc.edu/jobs-admissions/faculty-recruiting for details).

Applicants are encouraged to submit their applications along with supporting material as soon as possible, but no later than January 15, 2014 to ensure full consideration.

For further information, send mail to recruiting@cs.wisc.edu.

Wisconsin Institute for Discovery
Faculty Positions: Professors/Associate Professors/ Assistant Professors

The Wisconsin Institute for Discovery (WID) at the University (www.wid.wisc.edu) invites applications for faculty openings in Optimization and its Applications. Multiple opportunities are available at the Assistant, Associate, or Full Professor level. Successful candidates will occupy a new state-of-the-art and centrally-located WID research facility specifically designed to spark and support cross-disciplinary collaborations.

For specific details regarding the Optimization positions, see:

http://www.ohr.wisc.edu/WebListing/Unclassified/PVLSummary.aspx?pvlnum=77887 .

The University is an Equal Opportunity/Affirmative Action employer and encourages women and minorities to apply. Unless confidentiality is requested in writing, information regarding the applicants must be released on request. Finalists cannot be guaranteed confidentiality. A criminal background check may be conducted prior to hiring.

Computer Science
UNIVERSITY OF TORONTO

Assistant Professor - Computer Science

The **University of Toronto** invites applications for three tenure-stream appointments in Computer Science at the rank of Assistant Professor. The appointments will be with the tri-campus Graduate Department of Computer Science and its affiliated undergraduate departments:

- Department of Mathematical and Computational Sciences, University of Toronto Mississauga: specific areas of interest include operating systems, networks, distributed systems, database systems, computer architecture, programming languages, and software engineering.
- Department of Computer and Mathematical Sciences, University of Toronto Scarborough: all areas of Computer Science.
- Department of Computer Science, University of Toronto St. George: all areas of Computer Science that touch upon Big Data in the broadest possible sense-including but certainly not limited to theoretical foundations, algorithms, systems, software, applications and cross-disciplinary research. The candidate may also be nominated for a prestigious Tier II Canada Research Chair. For further information on these federally endowed Chairs see *www.chairs.gc.ca*.

All appointments will begin on July 1, 2014.

The University of Toronto is an international leader in computer science research and education. Successful candidates are expected to pursue innovative research at the highest level; to establish a strong, externally funded research program; to have a strong commitment to undergraduate and graduate teaching; to contribute to the enrichment of undergraduate programs in their department; and to participate actively in the Graduate Department of Computer Science.

Candidates should have a Ph.D. in computer science or a related field by the date of appointment or shortly thereafter. Evidence of excellence in teaching and research is required. Salaries are competitive with our North American peers and will be commensurate with qualifications and experience.

Applicants should apply online through AcademicJobsOnline, *https://academicjobsonline.org/ajo/jobs/3615*, and include a curriculum vitae, a list of publications, research and teaching statements, and the names and email addresses of at least three references, who will upload confidential letters.

To receive full consideration, applications should be received by January 10, 2014.

For more information about Computer Science on the three campuses of the University of Toronto, see our websites (St George: web.cs.toronto.edu; UTM: *www.utm.utoronto.ca/math-cs-stats*; UTSC: *www.utsc.utoronto.ca/cms*), or contact Sara Burns at *recruit@cs.toronto.edu*.

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. The University is responsive to the needs of dual career couples. The University of Toronto offers the opportunity to conduct research, teach, and live in one of the most diverse cities in the world. All qualified candidates are encouraged to apply; however, Canadians and permanent residents will be given priority.

Assistant Professor - Computational Biology

The **Department of Computer Science** and **the Donnelly Centre for Cellular and Biomolecular Research at the University of Toronto** invite applications for a tenure-stream appointment in Computational Biology or Bioinformatics. The appointment is at the rank of Assistant Professor and will begin on July 1, 2014.

We seek an emerging researcher with an exceptional research record and excellent teaching credentials. The successful candidate is expected to pursue innovative research at the highest level; to establish a strong, externally funded research program; to have a strong commitment to undergraduate and graduate teaching; and to participate actively in the Donnelly Centre and Department of Computer Science. Successful candidates will also have the opportunity to take advantage of the University's strengths in biology and bioinformatics—and computational, medical and biological sciences more broadly—and to facilitate further interaction with other units. To facilitate such interactions, the successful candidate will hold a joint appointment between the Department of Computer Science (51%) and the Donnelly Centre (49%). The candidate may also be nominated for a prestigious Tier II Canada Research Chair. For further information on these federally endowed Chairs see www.chairs.gc.ca.

The Department of Computer Science is an international leader in research and teaching, with recognized strength in most areas of computer science. The Donnelly Centre is an interdisciplinary research institute at the University of Toronto with the mandate to create a research environment that encourages integration of biology, computer science, engineering and chemistry, and that spans leading areas of biomedical research. Toronto is a vibrant and cosmopolitan city, one of the most desirable in the world in which to work and live, and a major centre for advanced computer, medical and biological technologies with strong ties to the University.

Applicants should apply online at *http://academicjobsonline.com/ajo/jobs/3641*, and include curriculum vitae, a list of publications, a research and teaching statement, and the names and email addresses of at least three references. Other supporting materials may also be included. Although we expect applicants to have a PhD and postdoctoral training in the computational sciences (computer science, computational biology and quantitative biology), exceptional candidates with recent or imminently-expected PhDs will be also considered.

Evidence of excellence in research and teaching is required. Salaries are competitive with our North American peers and will be commensurate with qualifications and experience.

To receive full consideration, applications should be received by January 10, 2014.

For more information on the Department of Computer Science see: www.cs.toronto.edu and for the Donnelly Centre for Cellular and Biomolecular Research see: *www.thedonnellycentre.utoronto.ca*. For questions regarding this position, please contact Sara Burns at *recruit@cs.toronto.edu*.

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.

Lecturer - Computer Science

The **Department of Computer Science, University of Toronto** seeks an enthusiastic and innovative teacher for a full-time teaching-stream appointment in the field of Computer Science. The appointment will be at the rank of Lecturer and will begin on July 1, 2014.

We seek candidates who have a record of excellent teaching, possess the intellectual curiosity to pursue novel and innovative methods of thinking and teaching, and are interested in establishing a long-term teaching career with the Department. Candidates from all areas of Computer Science are invited to apply. Particular attention will be given to candidates with an interest in or experience teaching Software Engineering. Candidates must have a graduate degree (PhD preferred) in computer science or a related field by the time of appointment or shortly thereafter. Responsibilities include undergraduate teaching, managing teaching assistants, developing course materials, and curriculum development. In addition, each faculty member has some responsibility for student recruitment and departmental administration.

Appointments at the rank of Lecturer may be renewed annually to a maximum of five years. In the fifth year of service, Lecturers shall be reviewed and a recommendation made with respect to promotion to the rank of Senior Lecturer. Senior Lecturers hold continuing appointments at the University.

Salary will be commensurate with qualifications and experience.

Applicants should apply online at *http://recruit.cs.toronto.edu/*, and include curriculum vitae, a list of publications, statement of career goals and teaching philosophy, teaching dossier, and the names and email addresses of three to five referees who may be contacted to provide a letter of reference. Review of applications will begin on January 15, 2014 and continue until the position is filled. If you have any questions regarding this position, please contact Sara Burns at *recruit@cs.toronto.edu*. The successful candidate will join a vibrant group of Lecturers who are engaged in pedagogical and curricular innovations, development of new teaching technologies, and research in computer science education. For more information about the Department of Computer Science, please visit our home page at www.cs.toronto.edu.

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.

[CONTINUED FROM P. 128] heard and we were here. Nearly everyone wanted to know more about the aliens: Did they have religion or music? How had they managed to ensure their own long-term survival?

But many people were wary of betraying our existence with a reply. International organizations assembled teams of the Earth's best and brightest, charging them with making our response sunny, informative, and, especially, non-threatening. This was an agonizing and protracted task. However, scores of top-flight academic researchers schooled in sociology, psychology, and even risk assessment put together a modern *New England Primer*, a condensed letter of introduction from humans to aliens, finally launching it skyward from the Arecibo antenna three years later. It was repeated 100 times, in case the aliens at the other end missed the beginning of the transmission.

The earthly public hunkered down in anticipation of a response. We were not alone. We were not a miracle. We were simply another bit of intelligence in a cosmos that was vast and old, and there was comfort in the thought we might learn from one another.

Surprisingly, and less than a month later, a second signal was detected, this time from a star system in the direction of Pegasus. SETI scientists had apparently stumbled on the cosmic "hailing frequency," and suddenly alien societies were turning up like crabgrass. The new transmission also contained an endlessly repeated message... a warning...

"Do not respond to the other [Cassiopeia] broadcast!"

Analysts pondered how the senders could know of the first transmission, and why they were telling us not to answer. Contact had produced a disturbing mystery.

Because the new signal was pictorial and highly redundant, its complete contents were relatively easy to decipher. Apparently many thousands of years ago, inhabitants of this second world had picked up the Cassiopeia broadcast. They too had replied, hoping to establish some rudimentary interspecies communication. But they also quickly uncovered a painfully unwelcome fact: The transmission was a ruse, a Trojan horse, emitted to serve the ends of some group whose description could be translated only as "galactic traders." They had long ago developed a strategy to tempt these societies to betray their presence.

Many reverted to primitive behavior, ranging from the hedonistic to the monastic, while others frantically pondered a fix.

Those that did—any responding world—were soon targeted by massive photonic weapons, enormous infrared lasers that could boil off a planet's atmosphere and incinerate much of its landmass.

Some people immediately questioned whether this second transmission was itself a hoax. But the Pegasus message explicitly said there was no point in any response to them. They were long gone. In their death throes, they had constructed a beacon to warn others.

Our situation was as plain as it was demoralizing. The caution from Pegasus was too late, and our unwitting invitation to earthly destruction was indeed en route at the speed of light.

Humanity seemed destined to stew and fret for the next 32 years, the time required for our transmission to reach them and their photon beam to travel to Earth. In our enthusiasm and endless curiosity we had lit a fuse, and it was now just a matter of waiting out the consequences. The people of Earth lamented that their children would have no grown children of their own. For those who cared about such things, it seemed possible that *Homo sapiens*' greatest discovery would also be its last.

Many reverted to primitive behavior, ranging from the hedonistic to the monastic; others frantically pondered a fix. Given 32 years, there might be time to evacuate at least a few thousand people to some hastily built orbiting space stations or perhaps a base on the Moon. But most took comfort in the fact that our message was carefully considered. Any truly advanced aliens would recognize we were neither malevolent nor harmful. Our reasoned diplomacy would be our salvation.

However, as these developments roiled society, a few scientists combined simple extrapolations with some even simpler calculations to reveal a startling new truth.

Any extraterrestrials able to construct interstellar weaponry would also have radio telescopes far more sensitive than our own. Yes, they could easily pick up our reply, now on its way. But decades before that message arrived, they would have detected many of the high-frequency transmissions sent willy-nilly into space since the Second World War. Those radar, television, and FM radio broadcasts had preceded our *Primer*, and could not be recalled. We had already alerted the aliens to our presence. Our carefully considered message was merely a coda to decades of human cacophony.

This dismaying realization was so troubling the researchers were reluctant to make it public. Some thought honest disclosure was the best policy, others that revealing this analysis would only further demoralize society, crippling any effort to construct livable refuges beyond Earth.

This discussion was quickly moot. The inhabitants of an unseen planet around a red dwarf star in the direction of Cassiopeia had already heard humanity's first radio noise, and had made their decision.

This is, of course, now history, possibly ancient history. Still, this brief testimony might be useful to you. Perhaps you can benefit from knowing that on an ordinary day four years after the first human discovery of cosmic intelligence, seven billion members of this planet's only sentient species looked to the sky, watching and wondering, as patches of the stratosphere began to darken, grow, and explode into flame. C

Seth Shostak (seth@seti.org) is the senior astronomer at the SETI Institute in Mountain View, CA.

From the intersection of computational science and technological speculation, with boundaries limited only by our ability to imagine what could be.

DOI:10.1145/2555933

Seth Shostak

Future Tense
The Second Signal

Even cosmic enlightenment can involve unwelcome contact.

WHEN THAT FIRST alien broadcast came in—no more than a thin stream of bits, really—it was just about the biggest news story ever.

It was also no surprise. For years, scientists had assured themselves, and every documentary TV maker who asked, that tuning in E.T. would be the greatest discovery ever. If researchers finally found proof of other intelligence lurking in our Milky Way galaxy, they would be shoo-ins for a Nobel Prize. More than that, the discovery would also—somehow—change everything.

Well, everything did change, on a scale far larger than predicted. However, it was because of the second signal, not the first...

The initial signal, picked up in a routine survey of red dwarf star systems by SETI, the Search for Extraterrestrial Intelligence, was only a short, radio ping from the direction of Cassiopeia. The code frame was roughly a minute long and repeated as long as the antennas continued to stare. The frame contained several kilobits of code, arranged in a pictorial matrix, clearly marking it as a deliberate effort to get our attention. Whatever brainpower authored the broadcast was succinct, sending the equivalent of a business card laced with a few astronomical facts about the aliens' home star system and planet. Earthly pundits argued the senders were likely establishing a data link before inundating us with zetabytes of information, possibly revealing some great truths about the universe.

That seemed reasonable. But everyone was surprised how close the transmitter was, a fact that made the data link feasible. Before the discovery, estimates of the number of technical civilizations in the galaxy ranged from thousands to millions. If true, societies should be separated by at least many hundreds of light-years, on average. But observations with antenna arrays indicated the source of the Cassiopeia transmission was a star system only 16 light-years away. Next door, really. SETI scientists admitted that inhabited worlds could be more commonplace than once believed. Another suggestion was we were, by chance, in an urbanized sector of the galaxy.

The short distance prompted a clamor for a reply, to tell these neighbors they were [CONTINUED ON P. 127]

PHOTOGRAPH BY SETH SHOSTAK/SETI INSTITUTE

Nearly everyone wanted to know more about the aliens: Did they have religion or music? How had they managed to ensure their long-term survival?

ACM
SIGIR 2014
JULY 6 – 11, 2014

THE 37TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE

Planning is well underway for the 37th Annual ACM SIGIR Conference, to be held on the Gold Coast, Queensland from Sunday 6 – Friday 11 July 2014.

SIGIR is the major international forum for the presentation of new research results and for the demonstration of new systems and techniques in the broad field of information retrieval. Next year's conference will feature 6 days of papers, posters, demonstrations, tutorials and workshops focused on research and development in the area of informational retrieval, also known as search.

The Conference and Program Chairs are now inviting all those working in areas related to information retrieval to submit original papers related to any aspect of information retrieval theory and foundation, techniques and application. A list of key submission dates, relevant paper topics, submission guidelines and instructions are now available on the official SIGIR 2014 Conference website: http://sigir.org/sigir2014/callforpapers.php. Abstract submission closes 20 January 2014.

In addition, to a full scientific program the conference presents delegates with the perfect networking opportunity, bringing together several hundred researchers, academic faculty, students and industry leaders from around the world.

SIGIR 2014 will take place at one of Australia's premier tourist destinations, the Gold Coast. From the iconic Surfers Paradise beach, to the sophisticated dining precincts of Broadbeach and out to the lush, green Hinterland, there is a new experience waiting for you at every turn on the Gold Coast. Theme parks, world-renowned beaches, shopping and almost year-round sunshine are just a few reasons why delegates will enjoy this vibrant coastal city.

From the SIGIR Conference Organising committee we hope to see you on the Gold Coast in 2014 for the 37th Annual ACM SIGIR Conference.

VEE 2014

10th ACM SIGPLAN/SIGOPS international conference on

Virtual Execution Environments

Salt Lake City 1–2 March 2014 with ASPLOS'14

Everything virtualization related, across all layers of the software stack down to the microarchitectural level

Call For Participation

General Chair
Martin Hirzel *(IBM Research)*

Program Committee
Remzi Arpaci-Dusseau *(UW Madison)*
David F. Bacon *(IBM Research)*
Muli Ben-Yehuda *(Technion & Stratoscale)*
Dilma Da Silva *(Qualcomm Research)*
Angela Demke Brown *(U of Toronto)*
David Dice *(Oracle)*
Ajay Gulati *(VMware)*
Sam Guyer *(Tufts U)*
Antony Hosking *(Purdue U)*

Program Co-chairs
Erez Petrank *(Technion)*
Dan Tsafrir *(Technion)*

Galen Hunt *(MSR)*
Doug Lea *(SUNY at Oswego)*
Gilles Muller *(INRIA)*
Todd Mytkowicz *(MSR)*
Mathias Payer *(UC Berkeley)*
Donald Porter *(Stony Brook U)*
Karsten Schwan *(Georgia Tech)*
Liuba Shrira *(Brandeis U)*
Bjarne Steensgaard *(Microsoft)*

http://vee2014.org

in cooperation with ***USENIX***