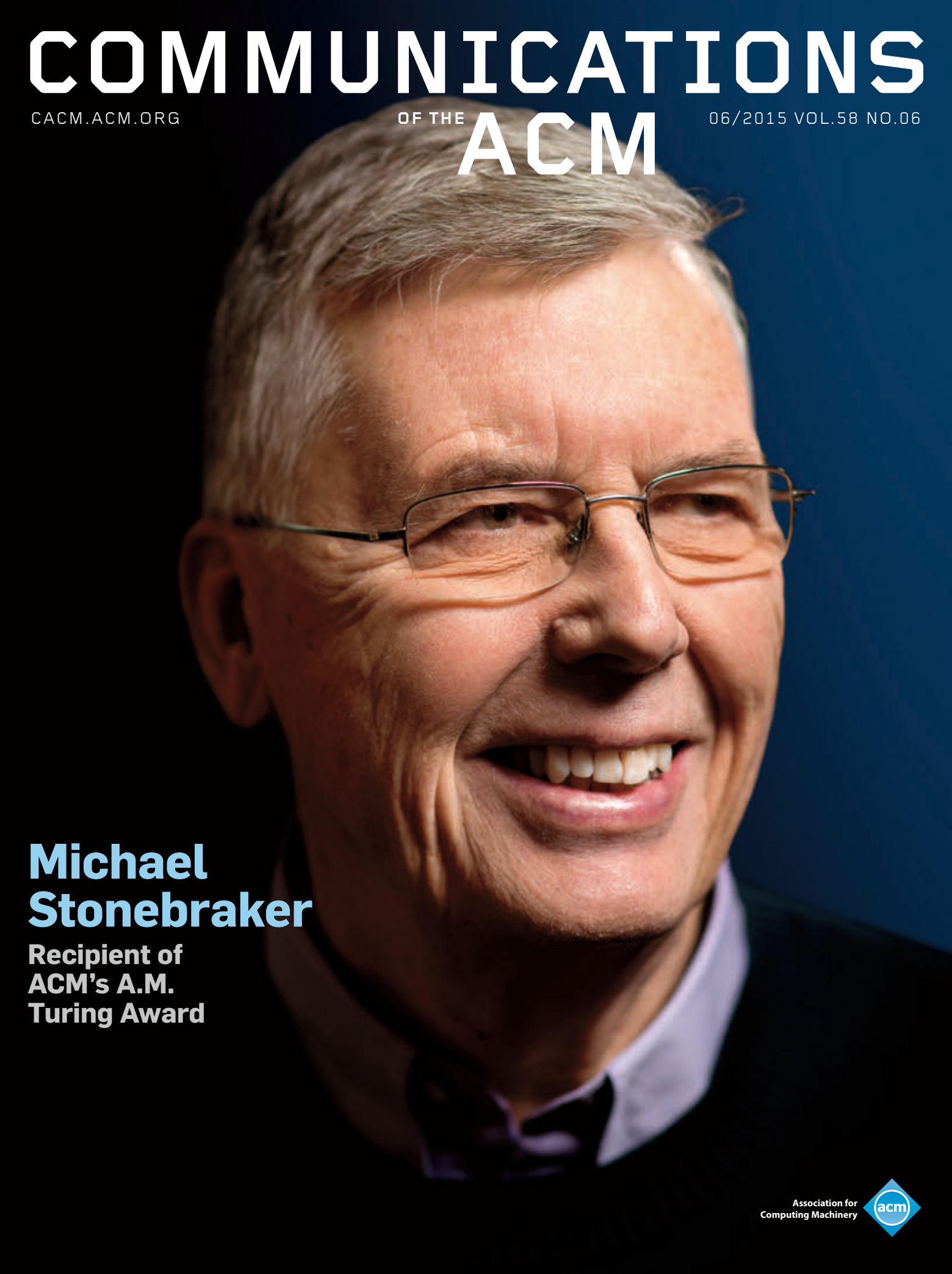
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

06/2015 VOL.58 NO.06



Michael Stonebraker

Recipient of ACM's A.M. Turing Award

Association for Computing Machinery

OMNITI PRESENTS

Surge '15



THE SCALABILITY CONFERENCE

Surge brings the best and brightest thought leaders in web operations together to share and learn about scalability and performance challenges of some of the largest infrastructures in the world.

TOPICS COVERED INCLUDE:

Scaling Infrastructure including configuration management, monitoring, hybrid cloud architectures, storage systems, and other challenges seen at scale.

Scaling Architecture including distributed databases, load balancing and resource allocation, decoupled and fault-tolerant systems.

Scaling Organizations including team structure, review techniques, code review processes, and establishing cross-organization accountability.

Don't miss out on an opportunity to discuss scalability matters ... because scalability matters.

surge.omniti.com

SEPTEMBER 23 - 25, 2015

Gaylord National Resort and Conference Center in Maryland.

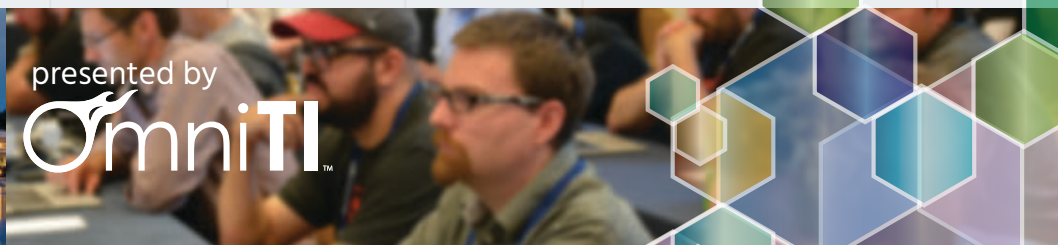
ACM Members Get

$100 OFF

Use code: ACM

visit surge.omniti.com





presented by OmniTI

THE ACM A. M. TURING AWARD

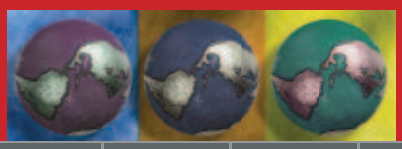
by the community ◆ from the community ◆ for the community



ACM and Google congratulate

MICHAEL STONEBRAKER

For fundamental contributions to the concepts and practices underlying modern database systems.



"The efficient and effective management of Big Data is crucial to our 21st century global economy," said Google Senior Vice President of Knowledge Alan Eustace. "Michael Stonebraker invented many of the architectures and strategies that are the foundation of virtually all modern database systems."

Alan Eustace
Vice President of Knowledge
Google Inc.

Google™

For more information see http://research.google.com/

Financial support for the ACM A. M. Turing Award is provided by Google Inc.

COMMUNICATIONS OF THE ACM

Departments

5 Farewell from ACM's CEO
Thank You...
By John White

7 Cerf's Up
A Celebration of Accomplishments
By Vinton G. Cerf

8 Letters to the Editor
To Learn CS Principles, Start in the Cafeteria Line

10 BLOG@CACM
Bringing Evidence-Based Education to CS
Mark Guzdial says computer science teachers should use more evidence, less intuition.

41 Calendar

101 Careers

Last Byte

104 Q&A
The Path to Clean Data
Michael Stonebraker on Ingres, Postgres, dividing his time between academia and start-ups, and why "one size fits none."
By Leah Hoffmann

Watch Stonebraker discuss his work in this exclusive *Communications* video.

News



12 **Plenty of Proteins**
The growth of structural biology brings new challenges for the world's protein data archive.
By Neil Savage

15 **Between the Lines**
Smartphone apps are driving changes in the way people park. Sensors, crowdsourcing, and big data are making it easier to find open parking spots.
By Samuel Greengard

18 **Estonia: A Model for e-Government**
Over the next decade, the population of Estonia is expected to soar more than 600% as the country becomes the first in the world to open its borders to an influx of e-residents.
By Gary Anthes

21 **Klaus Tschira: 1940–2015**
By Andreas Reuter

22 **Forging Relationships**
Michael Stonebraker didn't realize at the outset that it would take six years to create INGRES, one of the world's first relational databases.
By Neil Savage

Viewpoints

24 Inside Risks
Routing Money, Not Packets
Revisiting network neutrality.
By Vishal Misra

28 The Profession of IT
Emergent Innovation
Fernando Flores, president of Chile's National Innovation Council for Competitiveness, discusses a new common sense about innovation.
By Peter J. Denning

32 Kode Vicious
Lazarus Code
No one expects the Spanish Acquisition.
By George V. Neville-Neil

34 Viewpoint
Forgetting Made (Too) Easy
Considering the implications of digital data removal implementations.
By Meg Leta Jones

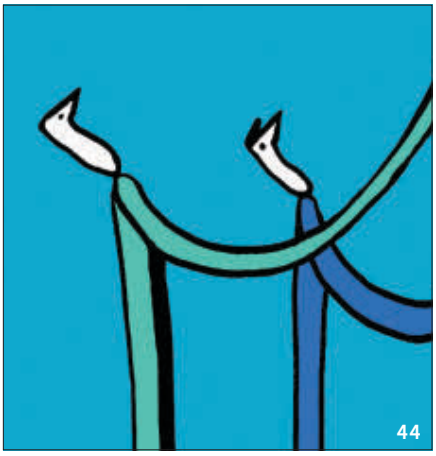
36 Viewpoint
Created Computed Universe
Computing crosses cosmology and makes the case for agnosticism.
By Yannis Papakonstantinou

39 Interview
An Interview with U.S. Chief Technology Officer Megan Smith
Recently appointed U.S. CTO Megan Smith discusses her evolving governmental role.
By Vinton G. Cerf

IMAGE BY JURGEN SCHULZE, UC-SAN DIEGO. COURTESY OF NATIONAL SCIENCE FOUNDATION

06/2015
VOL. 58 NO. 06

Practice

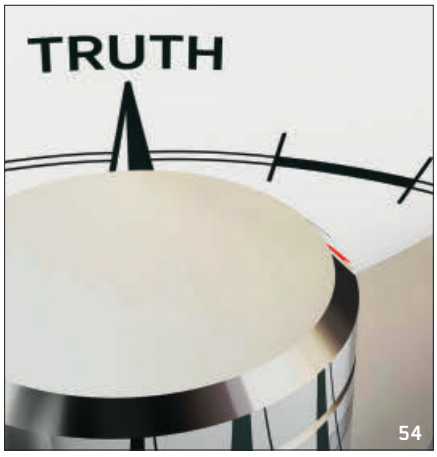


44 **The Science of Managing Data Science**
Lessons learned managing a data science research team.
By Kate Matsudaira

48 **Reliable Cron Across the Planet**
...or: How I stopped worrying and learned to love time.
By Štěpán Davidovič and Kavita Guliani

Articles' development led by acmqueue
queue.acm.org

Contributed Articles

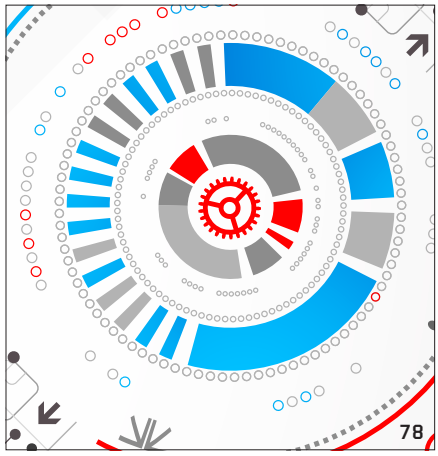


54 **Evaluation Without Ground Truth in Social Media Research**
Even without it, some ingenious methods can be developed to help verify users' social media behavioral patterns.
By Reza Zafarani and Huan Liu

61 **Colored Petri Nets: A Graphical Language for Formal Modeling and Validation of Concurrent Systems**
Formal executable models enable systematic evaluation of system designs prior to implementation and deployment.
By Kurt Jensen and Lars M. Kristensen

71 Turing Lecture
The Computer Science of Concurrency: The Early Years
Leslie Lamport is the recipient of the 2013 ACM A.M. Turing Award.
By Leslie Lamport

Review Articles



78 **Future Internets Escape the Simulator**
Future Internet testbeds permit experiments not possible in today's public Net or commercial cloud services.
By Mark Berman, Piet Demeester, Jae Woo Lee, Kiran Nagaraja, Michael Zink, Didier Colle, Dilip Kumar Krishnappa, Dipankar Raychaudhuri, Henning Schulzrinne, Ivan Seskar, and Sachin Sharma

Research Highlights

92 Technical Perspective
Virtual Reality in Your Living Room
By Patrick Baudisch

93 **IllumiRoom: Immersive Experiences Beyond the TV Screen**
By Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson

Watch the authors discuss this work in this exclusive *Communications* video.



About the Cover: Michael Stonebraker, recipient of ACM's 2014 A.M. Turing Award, pictured on the campus of Massachusetts Institute of Technology in Cambridge by Boston-based photographer Jared Leeds. Read about the origins of INGRES and POSTGRES on p. 22 and Stonebraker's challenges juggling academics and start-ups on p. 104.

IMAGES BY JUSTINE BECKETT; DONSCARPO MICROSTOCK; OMELCHENKO

Association for Computing Machinery
Advancing Computing as a Science & Profession

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.

Executive Director and CEO
John White
Deputy Executive Director and COO
Patricia Ryan
Director, Office of Information Systems
Wayne Graves
Director, Office of Financial Services
Darren Ramdin
Director, Office of SIG Services
Donna Cappo
Director, Office of Publications
Bernard Rous
Director, Office of Group Publishing
Scott E. Delman

ACM COUNCIL

President
Alexander L. Wolf
Vice-President
Vicki L. Hanson
Secretary/Treasurer
Erik Altman
Past President
Vinton G. Cerf
Chair, SGB Board
Patrick Madden
Co-Chairs, Publications Board
Jack Davidson and Joseph Konstan
Members-at-Large
Eric Allman; Ricardo Baeza-Yates; Cherri Pancake; Radia Perlman; Mary Lou Soffa; Eugene Spafford; Per Stenström
SGB Council Representatives
Paul Beame; Barbara Boucher Owens; Andrew Sears

BOARD CHAIRS

Education Board
Mehran Sahami and Jane Chu Prey
Practitioners Board
George Neville-Neil

REGIONAL COUNCIL CHAIRS

ACM Europe Council
Fabrizio Gagliardi
ACM India Council
Srinivas Padmanabhuni
ACM China Council
Jiaguang Sun

PUBLICATIONS BOARD

Co-Chairs
Jack Davidson; Joseph Konstan
Board Members
Ronald F. Boisvert; Nikil Dutt; Roch Guerrin; Carol Hutchins; Yannis Ioannidis; Catherine McGeoch; M. Tamer Ozsu; Mary Lou Soffa

ACM U.S. Public Policy Office
Renee Dopplick, Director
1828 L Street, N.W., Suite 800
Washington, DC 20036 USA
T (202) 659-9711; F (202) 667-1066

Computer Science Teachers Association
Lissa Clayborn, Acting Executive Director

STAFF

DIRECTOR OF GROUP PUBLISHING
Scott E. Delman
cacm-publisher@cacm.acm.org

Executive Editor
Diane Crawford
Managing Editor
Thomas E. Lambert
Senior Editor
Andrew Rosenbloom
Senior Editor/News
Larry Fisher
Web Editor
David Roman
Rights and Permissions
Deborah Cotton

Art Director
Andrij Borys
Associate Art Director
Margaret Gray
Assistant Art Director
Mia Angelica Balaquiot
Designer
Iwona Usakiewicz
Production Manager
Lynn D'Addesio
Director of Media Sales
Jennifer Ruzicka
Public Relations Coordinator
Virginia Gold
Publications Assistant
Juliet Chance

Columnists
David Anderson; Phillip G. Armour; Michael Cusumano; Peter J. Denning; Mark Guzdial; Thomas Haigh; Leah Hoffmann; Mari Sako; Pamela Samuelson; Marshall Van Alstyne

CONTACT POINTS

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

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/

ACM ADVERTISING DEPARTMENT

2 Penn Plaza, Suite 701, New York, NY 10121-0701
T (212) 626-0686
F (212) 869-0481

Director of Media Sales
Jennifer Ruzicka
jen.ruzicka@hq.acm.org

Media Kit acmmediasales@acm.org

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

EDITORIAL BOARD

EDITOR-IN-CHIEF
Moshe Y. Vardi
eic@cacm.acm.org

NEWS
Co-Chairs
William Pulleyblank and Marc Snir
Board Members
Mei Kobayashi; Kurt Mehlhorn; Michael Mitzenmacher; Rajeev Rastogi

VIEWPOINTS
Co-Chairs
Tim Finin; Susanne E. Hambrusch; John Leslie King
Board Members
William Aspray; Stefan Bechtold; Michael L. Best; Judith Bishop; Stuart I. Feldman; Peter Freeman; Mark Guzdial; Rachelle Hollander; Richard Ladner; Carl Landwehr; Carlos Jose Pereira de Lucena; Beng Chin Ooi; Loren Terveen; Marshall Van Alstyne; Jeannette Wing

PRACTICE
Co-Chairs
Stephen Bourne
Board Members
Eric Allman; Charles Beeler; Bryan Cantrill; Terry Coatta; Stuart Feldman; Benjamin Fried; Pat Hanrahan; Tom Limoncelli; Kate Matsudaira; Marshall Kirk McKusick; Erik Meijer; George Neville-Neil; Theo Schlossnagle; Jim Waldo

The Practice section of the CACM Editorial Board also serves as the Editorial Board of acmqueue.

CONTRIBUTED ARTICLES
Co-Chairs
Al Aho and Andrew Chien
Board Members
William Aiello; Robert Austin; Elisa Bertino; Gilles Brassard; Kim Bruce; Alan Bundy; Peter Buneman; Peter Druschel; Carlo Ghezzi; Carl Gutwin; Gal A. Kaminka; James Larus; Igor Markov; Gail C. Murphy; Bernhard Nebel; Lionel M. Ni; Kenton O'Hara; Sriram Rajamani; Marie-Christine Rousset; Avi Rubin; Krishan Sabnani; Ron Shamir; Yoav Shoham; Larry Snyder; Michael Vitale; Wolfgang Wahlster; Hannes Werthner; Reinhard Wilhelm

RESEARCH HIGHLIGHTS
Co-Chairs
Azer Bestovros and Gregory Morrisett
Board Members
Martin Abadi; Amr El Abbadi; Sanjeev Arora; Dan Boneh; Andrei Broder; Doug Burger; Stuart K. Card; Jeff Chase; Jon Crowcroft; Sandhya Dwaekadas; Matt Dwyer; Alon Halevy; Maurice Herlihy; Norm Jouppi; Andrew B. Kahng; Henry Kautz; Xavier Leroy; Kobbi Nissim; Mendel Rosenblum; David Salesin; Steve Seitz; Guy Steele, Jr.; David Wagner; Margaret H. Wright

WEB
Chair
James Landay
Board Members
Marti Hearst; Jason I. Hong; Jeff Johnson; Wendy E. MacKay

ACM Copyright Notice
Copyright © 2015 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from permissions@acm.org or fax (212) 869-0481.

For other copying of articles that carry a code at the bottom of the first or last page or screen display, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center; www.copyright.com.

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:10.1145/2767179

John White

Thank You...

After almost 17 years, I am retiring and stepping down as ACM CEO. My term as CEO combined with 17 years as a volunteer means ACM has been a huge part of my professional life. I appreciate this opportunity to reflect a bit on these 34 years, and to say thank you.

My ACM involvement began when I was a professor of computer science at the University of Connecticut in the mid-1970s and was elected Secretary/Treasurer of SIGPLAN. The fact that ACM was an association where volunteer leadership positions were generally determined by open elections impressed me and has, I believe, served ACM well. Beginning with this SIGPLAN position, over the next 17 years I served as Vice Chair and Chair of SIGPLAN, Northwest Regional Representative to Council, SIG Board Member, SIG Board Chair, ACM Vice President, and ACM President. These years were at times challenging. ACM was not as financially healthy as it is today. Council was a large (26 members) and somewhat unruly body. There was a real division between the technical/SIG community and the general activities of ACM.

Despite these challenges, many substantive changes were made over this period. Council was reduced to 16 members, the Executive Committee began to really own Council meetings and ensure Council focused on strategic issues, we launched the SIG Project Fund to support general ACM activities, our policy effort (USACM) was launched, our push to see more women enter and stay in computing (ACM-W) was initiated, and our first efforts in K–12 computer science education started.

These changes began to bring together the myriad parts of ACM. What made it all work was the incredible commitment of senior volunteers (and the inherent support of their institutions) and ACM staff. For me, the support I received from my boss and cohort in running the Computer Science Lab at Xerox PARC, Mark Weiser, and the head of PARC, John Seely Brown, to take the time necessary to run ACM when I was president was remarkable.

I left PARC to become CEO of ACM in November of 1998. At the time, the Digital Library (DL) was moving from a "free" experiment for members to a real product and service. What happened over the next few years was extraordinary. Between the senior staff at ACM HQ (Wayne Graves, Bernie Rous, Pat Ryan) and senior volunteers (particularly SIG and Publications Board leaders) we built a vision for the DL as a state-of-the-art, richly linked, digital collection of everything ACM had ever published embedded in a bibliographic database that defined computing. The launch of this new DL in 2001 set the stage for a remarkable shift in ACM. The DL became the centerpiece of the association. Publishing revenues and revenue sharing with the SIGs grew substantially. The divisions that had hindered ACM over the preceding two decades faded.

The solid financial base enabled by the DL allowed ACM to take on many new challenges. Foremost among these was the decision to invest in being a truly international organization. The effort started during the presidencies of Dave Patterson and Stu Feldman, but really grew under the leadership of Wendy Hall. ACM now has a solid presence in Europe, India, and China. Membership growth is coming from outside the U.S., and U.S. members are no longer the majority of ACM members. We are an international organization.

There have been many other areas of growth and development I am proud of, including our work related to diversity and gender equity, K–12 computer science education, public policy, education policy, and our ongoing efforts to offer publishing options that meet the needs of the ACM community.

Despite being in a very strong position as an association, ACM faces challenges. Our publishing models must continue evolving in the face of ongoing pressure for increased free access to published information. The fundamental notion of a membership organization is being challenged by the new world of free, Internet-based communities and services. That said, ACM has always been able to tap into the best thinking for meeting challenges, and I believe it will continue to do so under the leadership of the next CEO—an individual who knows the community well and has an incredible record of accomplishments. ACM is in good hands.

It has been a pleasure to work with hundreds of volunteers over the decades. I am particularly grateful to the ACM presidents I have worked with as CEO: Barbara Simons, Steve Bourne, Maria Klawe, Dave Patterson, Stu Feldman, Wendy Hall, Alain Chesnais, Vint Cerf, and Alex Wolf.

The hardest good-bye and the biggest "thank you," however, goes to the dedicated, hardworking, capable, and insightful staff of ACM: the directors, the managers, the administrative staff—everyone! And here there is one very special thank you—and that is to Pat Ryan, ACM COO. Pat has been my partner in everything I have tackled at ACM and in everything ACM has accomplished. She is a remarkable manager and leader. She knows ACM better than anyone, she has amazing instincts, and she has been an absolute pleasure to work with. Thank you, Pat.

Serving as ACM CEO has been the highlight of my career and a privilege. Thank you.

© 2015 ACM 0001-0782/15/06 $15.00

SHAPE THE FUTURE OF COMPUTING.
JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits and resources that can advance your career and enrich your knowledge. We dare to be the best we can be, believing what we do is a force for good, and in joining together to shape the future of computing.

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

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

ACM STUDENT MEMBERSHIP:

- ❑ Student Membership: $19 USD
- ❑ Student Membership plus ACM Digital Library: $42 USD
- ❑ Student Membership plus Print *CACM* Magazine: $42 USD
- ❑ Student Membership with ACM Digital Library plus Print *CACM* Magazine: $62 USD

❑ **Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in all aspects of the computing field. Available at no additional cost.

Priority Code: CAPP

Payment Information

Name

ACM Member #

Mailing Address

City/State/Province

ZIP/Postal Code/Country

Email

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc., in U.S. dollars or equivalent in foreign currency.

❑ AMEX ❑ VISA/MasterCard ❑ Check/money order

Total Amount Due

Credit Card #

Exp. Date

Signature

Purposes of ACM

ACM is dedicated to:

1) Advancing the art, science, engineering, and application of information technology
2) Fostering the open interchange of information to serve both professionals and the public
3) Promoting the highest professional and ethics standards

Return completed application to:
ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

Satisfaction Guaranteed!

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

Association for Computing Machinery

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)

Hours: 8:30AM - 4:30PM (US EST)
Fax: 212-944-1318

acmhelp@acm.org
acm.org/join/CAPP



DOI:10.1145/2766461

Vinton G. Cerf

A Celebration of Accomplishments

It is that time of year when ACM once again celebrates the accomplishments of an impressive array of contributors to our field, profession, and to ACM. The annual ACM Awards Gala takes place on June 20 in San Francisco and this year, for the first time, the Turing Award recipient will receive $1,000,000 thanks to the generosity of Google.

Michael Stonebraker has been a fixture in the database constellation for many years and his contributions have resulted in the formation of new companies and new insights into database design and implementation. His work has been widely recognized and it is deeply satisfying to see it honored by ACM in this way. I am going to miss this celebration, owing to some required travel, so I would also like to extend my congratulations and well wishes to all of the honorees at the awards banquet this June.

As many of you may know, ACM's CEO John White announced his intention to retire from ACM last year and this milestone will be passed later in 2015. John has been a guiding star for ACM for decades, having served in many capacities including president from 1990–1992 and as the CEO since January 1999. It is impossible to summarize the far-reaching and constructive impact of John's tenure but our organization and our profession would be poorer without his passionate commitment to excellence, computing, and ACM. I have had the satisfaction of working with him during his long association with ACM and have admired his skill, diplomacy, creative instinct, and management talent. The more I spend time with him, the more I hope a little of that will rub off on me. He has a self-effacing style that proves an old Washington (D.C.) saying that you can accomplish a great deal if you don't care who gets credit for it. John has gotten *a lot* done under the many hats he has worn while serving ACM. Speaking as a former president and current council member, I can confirm he has built and sustained a well-functioning team at ACM, led by the COO, Pat Ryan. John and Pat have fashioned an enabling environment for volunteers and a solid team of full-time staff without whom ACM would be greatly diminished. I wish John well as he approaches this milestone and thank him for many years of faithful and effective leadership.

I would also like to take a moment to share something I learned during a recent visit to the U.S. Military Academy (USMA) at West Point. I was visiting the computer science and electrical engineering departments for the day and had a chance to meet with senior leadership, faculty members, and students. There was a significant fraction of women enrolled in these computer science and electrical engineering departments and I found them particularly energized. Having heard some sad tales of women in other university settings who felt unwelcomed in classrooms filled with males, some of whom lacked any ability to engage in civil discourse, I was surprised and pleased to hear that things seemed to be better at USMA. On enquiry, I learned one reason for this is the head of the department insisted on trying to make sure there were at least three women in each class. It turns out it is far easier and more effective to defend someone else than it is to defend yourself. The latter may be seen as self-serving while the former is often seen as reinforcing and justifiable. While this is by no means a sudden panacea for solving the inclusiveness problem, it struck me as a very effective tactic that might prove useful in other settings. Another instructor told me he had required men and women on the USMA crew (rowing team) to train in mixed teams to reinforce respect for the talents and abilities of both groups. These practical steps reinforced my hopes that we will find our way to serious improvement in the statistics of the participation of women in computer science and electrical engineering.

John White has gotten *a lot* done under the many hats he has worn while serving ACM.

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

DOI:10.1145/2764994

To Learn CS Principles, Start in the Cafeteria Line

I enjoyed Thomas J. Cortina's Broadening Participation Viewpoint "Reaching a Broader Population of Students through 'Unplugged' Activities" (Mar. 2015) on teaching students computer science unplugged, that is, without a computer, especially the "parity bit" game puzzle, but would like to suggest yet another way to show students computer science principles in action, by leading them through their school cafeteria.

Many computer science textbooks explore the concept of a last-in first-out stack by comparing it to the stack of plates in the cafeteria. The plate most recently pushed onto the stack is always the next one taken off the top. Only the topmost plate is visible and accessible, as a spring platform causes the other plates to sink below counter level. Yet the plate-stack analogy is only the tip of the iceberg when it comes to computer science principles; for example, a queue of diners waits to be served, first in, first out, so after the person in front is served, each other waiting person is promoted to the next position in the queue. The conveyor-toaster demonstrates pipelining and the multi-slice toaster batch processing. When preparing, say, a bacon cheeseburger, rules of parallel processing apply. The grill cook can fry the bacon and grill a beef patty in parallel but must wait for both processes to complete before assembling the cheeseburger, a serial step.

A cafeteria requiring shirts and shoes for service demonstrates Boolean logic. In fact, "Shirt and shoes required for service" and "No shirt, no shoes, no service" are equivalent due to De Morgan's laws and the rules of inference. A student sitting at a table to eat is like a software process allocating and occupying memory. When students clear their table after eating to let someone else sit down, they are freeing memory.

The various levels of food storage are like different levels of data storage. Food readily accessible in the steam trays behind the counter is like RAM cache. Food waiting in the kitchen is like space on a hard drive. And food in a freezer is like tape storage or a compressed file.

Likewise, the cafeteria suggestion box reflects an asymmetric encryption scheme. The slot for inserting suggestions is, in essence, public-key encryption, since, in theory, no one else can peek at a suggestion once it is inside the box. The key to unlocking the suggestion box is a private decryption key.

Students can see all sorts of computer science theories, metaphors, and parallels in non-computing environments, quenching their thirst for knowledge and understanding even as they satisfy their hunger for, say, food, too.

Fred Murphy, Mountain View, CA

Turing Was No Gauss

Maarten Bullynck et al.'s comparison of Alan M. Turing and Carl Friedrich Gauss in their Viewpoint "Why Did Computer Science Make a Hero Out of Turing?" (Mar. 2015) reminded me of a delightful spoof article written in 1948 called "A Very Short History of Mathematics" by W.W.O. Schlesinger and A.R. Curtis of the University of Cambridge who read it at a meeting of the University's mathematical society, then published it in *Eureka*, the society's journal.

The article ran little more than two pages, but German mathematician Gauss merited a full paragraph, reading in part: "Gauss invented so many things that it just isn't true ... In fact, whenever anyone invented anything in the first half of the nineteenth century, Gauss had invented it twenty years earlier, and was still alive to tell him so ..."

It is the world's loss that Turing died so young, though it is difficult to imagine him publicly saying to any younger computer scientist "I was there before you," even if he had lived, like Gauss, to the age of 77. In terms of personal modesty at least, the two seem worlds apart.

Stephen Robertson, London, U.K.

When Software Is Not Protected by Copyright

Pamela Samuelson's Legally Speaking Viewpoint "Copyrightability of Java APIs Revisited" (Mar. 2015), about the *Oracle v. Google* case now being considered for review by the U.S. Supreme Court, helpfully referred to certain considerations used in the Second Circuit's influential *Computer Associates v. Altai* decision to determine when an element of a computer program is not protected by copyright but understated the significance of the time period to which those considerations are applied.

One such consideration, Samuelson emphasized, is whether the element was "dictated by external factors" (such as "compatibility requirements of other programs"), suggesting such a compatibility or interoperability requirement favors a holding that the Java APIs at issue are not protected. However, the purpose of *Altai*'s "filtration" step, in which compatibility is one consideration, is to determine the scope of a program creator's copyright from the outset by filtering out unprotectable program elements, so the step is applied as of the time of the program's creation. At the time of creation of the software at issue in *Oracle v. Google*, the Java APIs were new, and their creation was not dictated by compatibility requirements of other programs, so this consideration argues in favor of, not against, their protection by copyright.

Under current case law, the need for compatibility/interoperability with currently popular programs and libraries remains a relevant factor in determining infringement, but as part of a fair use determination (such as was carried out in the *Sega v. Accolade* case Samuelson discussed), not as part of deciding which elements of a plaintiff's program are protected by copyright at all. The Federal Circuit decision on appeal remanded the case to the district court for a new trial on fair use.

It is, however, also reasonable to ask why, if interoperability is an important

enough factor in some cases to keep elements of a program from getting any copyright protection at all, that factor should be limited to interoperability at the time of the program's creation. In light of the developments in software and programming that have made interoperability a practical necessity in many cases, it may be time for the Supreme Court to consider whether factors held relevant to protectability in *Altai*, including, but not necessarily limited to, interoperability, should be considered as of the time of alleged infringement, as well as the time of program creation, without the need for a full-blown fair use trial.

Allowing the infringeability of elements of a program to change due to developments subsequent to its creation would be a significant modification of copyright case law on its face but may represent the best accommodation of the statutory language and principles of copyright to a type of work—computer software—in which functionality plays a far greater role than in the literary works upon which the existing doctrine is based, and for which externalities like interoperability and changes in those externalities may substantially increase the benefits of copyright protection to program creators.

Marc K. Temin, Newton, MA

Author's Response:

The Altai *decision did not say one way or the other whether constraints on the plaintiff's interface design had to exist at the time of creation of that software. Subsequent cases, most notably* Sega v. Accolade, *involved interfaces (such as for the Genesis platform) that could have been designed many ways initially by Sega. But when Accolade wanted to modify a videogame to run on the Genesis platform, its design decisions were constrained by the Sega interface if it was to achieve interoperability. The Ninth Circuit ruled the Sega interface was unprotectable by copyright law. The Federal Circuit was supposed to follow this Ninth Circuit precedent in* Oracle v. Google *but chose instead to misconstrue the case.*

Pamela Samuelson, Berkeley, CA

Give the Client a Working Model

Leslie Lamport's important Viewpoint "Who Builds a House without Drawing Blueprints?" (Apr. 2015) cited an eternal truth about the practice and craft of software engineering: that thought and planning must precede construction. But this analogy to architectural blueprints can be extended for the benefit of practitioner and client alike.

Designers of our built environment use blueprints and their modern digital equivalents to do much more than specify construction details for the trades. Models and blueprints are a communications tool that lets a client visualize and understand an architect's proposed design while allowing the client to guide the design. However, a model that serves as a requirements-refinement tool is often missing from software-construction practice, at least in my experience. For most projects, the client is given nothing that yields meaningful understanding of a system's behavior, due mainly to the extraordinary cost of providing a working model in advance of the system itself.

Lamport recommended mathematics as the best "language" for specifications. Yet most systems clients know less about formal logic than they do about probability, or little or nothing. Software utility to the client, outside the real-time domain, depends less on the precision illustrated by Lamport's text-formatting example than on the business functions important to the client. Which of us would notice if a built bedroom was an inch smaller than the architect's plan but would indeed consider legal action if the windows did not open?

Barry Needalman, Framingham, MA

Give Me the 5% Capable of Incisive Critique

Though Muaz A. Niazi's Letter to the Editor "The Case of the Missing Skillset" (Apr. 2015) made a valid point concerning the "lack of the skillsets needed to write efficient and verifiable parallel programs," the problem of "how to bridge the gap between the software industry and the community of formal-methods practitioners" is not as daunting as he suggested. Employers do not need 100% of their software developers to be able to write the kinds of code Niazi envisioned. All they need is a critical attitude and a tool that tells them whether a program or combination of programs violates established rules. Just as many playwrights can learn from a single incisive critic, business people need only 5% of the workforce at the level of expert critic. Many of the other software developers will learn from interacting with them over time, while employers learn which 15% are no longer needed.

Jack Ring, Gilbert, AZ

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.

© 2015 ACM 0001-0782/15/06 $15.00

Coming Next Month in COMMUNICATIONS

Exascale Computer and Big Data: The Next Frontier

Recent Developments in Unifying Logic and Probability

Using Free and Open Source Tools to Manage Software Quality

Low-Latency Distributed Application in Finance

The Past, Present, and Future of Password-Based Authentication on the Web

The Future of Computer Science and Engineering Is in Your Hands

Using Rhetorical Structure in Sentiment Analysis

Cache Efficient Functional Algorithms

Plus the latest news about deep learning applications, urban information systems, and state-of-the-art open source programming tools.

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:10.1145/2754947 http://cacm.acm.org/blogs/blog-cacm

Bringing Evidence-Based Education to CS

Mark Guzdial says computer science teachers should use more evidence, less intuition.



Mark Guzdial

Computing Education Must Go Beyond Intuition: The Need for Evidence-Based Practice

http://bit.ly/1DdxT3a

February 22, 2015

The highest-quality education practice is *evidence-based education*, defined by the U.S. Department of Education (http://1.usa.gov/1CWNB4o) as "the integration of professional wisdom with the best available empirical evidence in making decisions about how to deliver instruction." The movement toward evidence-based practice has swept through medicine (http://1.usa.gov/1IttwlY), psychology (http://bit.ly/1CBcRM2), and other fields.

Computing educators' practice would dramatically improve if we drew on evidence rather than intuition. Raymond Lister has been writing for years about the problem that computing education practitioners do not engage with evidence and the research on education (http://bit.ly/1EMqBY1). Lister calls the knowledge computing educators use in making teaching decisions "folk pedagogy" (http://bit.ly/1NL9xmq).

Folk pedagogy encourages what is believed to be best practice, but cannot validate best practice.

We have evidence computing teachers do not use evidence. Davide Fossati and I studied 14 CS teachers from three institutions (http://bit.ly/1BV9uNo). Fossati asked them about times they made a change in their teaching practice; why did they make the change, and how did they know if it was successful or not. They used intuition, informal discussion with students, and anecdotes. *Not a single teacher* used evidence such as class performance on a test or homework.

Without evidence, teachers rely on intuition informed by experience. Sometimes that intuition may be informed by years of experience. Sometimes that experience is not at all relevant.

The March 2015 issue of *Inroads* (http://bit.ly/1ItulLy) has a section on "The role of programming in a non-major, CS course." That is a topic of great interest to me, since I have been developing and studying a programming-based approach to teaching introductory computing we have used with non-CS majors at Georgia Tech for over 10 years (http://bit.ly/1AkpH2x). The article (http://bit.ly/1NLcoff) by Richard Kick and Frances Trees describes the AP CS Principles course, its research-based development, and experience at the pilot sites. The article by Steve Cooper and Wanda Dann (http://bit.ly/19ND6ok) presents findings from early research on teaching programming to children, and evidence from courses at Stanford and CMU using Alice. The other papers use no evidence.

Henry Walker's article (http://bit.ly/1EMkm6s) argues programming should not be a priority for a non-CS majors course:

Development of student problem-solving skills (with programming) requires time, so the inclusion of programming within a non-CS majors CS course comes at a price: significant other topics must be dropped.

The claim "significant other topics must be dropped" is empirical. Since there are many non-CS majors CS courses with programming described in the ACM literature, one could consider those courses and identify the significant topics dropped in favor of programming.

Henry does not consider these contrasting cases, but offers recommendations of what a non-major student *ought* to know about CS. Are CS faculty the right ones to define the learning goals for students who *do not* plan a career in computer science? When we designed Media Computation, we formed an advisory board of non-CS faculty to tell us what their students needed to know about computing. For example, a professor in architecture wanted his students to understand the difference between Photoshop and a CAD tool for manipulating a diagram. While both can "extend a line,"

Photoshop manipulates pixels, while CAD tools manipulate a vector representation that can be used to control automated devices like milling machines. I had not realized that was an important data representation learning objective for architecture.

Michael Goldweber makes two arguments in his paper against including programming in a non-CS majors course (http://bit.ly/1P3od1X). His first is like Walker's, in that he offers what *should* be in a non-CS majors course. He suggests an "*embarrassment* model of course and curriculum development" in which "one enumerates the topics for which, if their students did not know about/have experience with, one should feel embarrassment":

I assert this topics list does not include programming, not because programming is without merit, but because the inclusion of programming requires a time commitment that leaves an insufficient quantity of time for the key "embarrassment" topics. These include algorithmic problem solving, the use of abstraction to tame complexity, and the limits of computation. It is my position that it would be an embarrassment if a student coming out of a non-CS majors course could successfully program a solution to Selection Sort or write a program that draws a fractal tree, but have no idea of how to apply algorithmic problem-solving to real-world problems.

Goldweber suggests a real-world problem he believes non-CS majors should be able to solve after an introductory course to CS:

Given a graph representing cities and connecting highways, some of the cities house a Red Cross warehouse while one other city experiences a disaster; describe an algorithm for locating the closest Red Cross warehouse.

My suspicion based on research evidence is few CS majors or non-CS majors would be able to solve this problem, even after several classes (with or without programming). Transfer of knowledge is difficult to achieve, and students are particularly challenged to recognize a real-world problem is related to other knowledge they have learned (http://bit.ly/1BVbKnW). In the decades of studies that have tried to find such transfer, the research evidence is that computing courses do not help students develop general problem-solving skills (http://bit.ly/19NPK6O). That is just my suspicion. We could gather evidence to see if students get closer to achieving Goldweber's goal with a programming course or a non-programming course. He does not offer any evidence.

Goldweber's second argument is about the cost of programming in time and tedium. He dismisses the use of languages such as "Pascal, Java, Python" in which students engage in "wrestling matches with compilers over a missing or misplaced semicolon or squirrely bracket." He disparages graphical programming languages ("Scratch, Alice, Kodu") as "children's introduction to programming":

Should the learning outcomes for a child's introduction to the field be what we want for one's singular university-level non-major's course?

Betsy DiSalvo published a paper on a study of African-American teens who learned both Python and Alice (http://bit.ly/19MBo6u). There was not a clear winner. DiSalvo found the preference depended on the *careers* these students desired. Those interested in computer science as a career preferred Python. Those interested in media and design preferred Alice because they liked what they created, but also because the graphical nature made the high-level structure more evident. What one participant told DiSalvo sounds much like what Goldweber and Walker want students to achieve:

I like the top-down design (in Alice projects). We are able to break down the bigger problems into smaller problems and then even smaller problems, so it's a simpler way to file through. I think you could probably apply top-down design to anything in life.

The biggest concern I have for making education decisions without evidence is *who* is making the educational decisions *for whom*. Most CS faculty (reflected in the authorship of these articles) are white and male. Because we are CS faculty, most of our experience is rooted in being students in STEM fields.

Our intuition is likely wrong about non-CS majors. Most non-CS majors are not students in STEM. Non-CS majors are more diverse. We do not know their experience, goals, or desired careers. We should not assume we know how to teach them. We should not assume what non-CS majors need to know. When we want to know what CS majors will need in their careers, we ask industry advisors (as the CS2013 curriculum process did, http://bit.ly/1yHg0Gc). We know much less what non-CS majors might need to know in their careers. We need to gather evidence to determine what non-CS majors need to know about computing.

We need the humility to recognize what we do not know, and we need evidence to inform our decisions. You do not want your surgeon to apply the best practices of folk medicine on you. Our students deserve the best educational practices informed by evidence, not folk pedagogy informed by intuition.

Comments

Mark, I agree completely with your main points: CS educators should do more to base our practice on evidence rather than intuition, because that intuition is likely wrong. I would go further in several ways.

First, our intuition is likely wrong about CS and STEM majors too, even if we think we know what they need to know, since most of our students are unlike us.

Second, we should leverage evidence-based practices that give us more direct evidence about our students as they learn. For example, approaches like process-oriented guided inquiry learning (POGIL) and peer instruction help us observe students as they grapple with new ideas, so we can respond more quickly to problems. When my students work through a POGIL activity, I see where they need help, I can respond immediately, and I know what to revise for the future. It is easy for lecturers and their students to overestimate what has been "learned" until it has to be applied.

On the other hand, I am not sure how far behind we are in CS education; folk pedagogy still seems to be the norm on many campuses.

—Clif Kussmaul

Completely agree, Clif! I'm a huge fan of Peer Instruction (http://bit.ly/1y47mae) because it gives me evidence about my class, what they know and what they do not.

Carl Wieman reports over 70% of physics teachers are familiar with Physics Education Research results and use at least one finding in their teaching. We in CS Ed are not there yet.

—Mark Guzdial

Walker and Goldweber have written a response to this post on the ACM Inroads *blog, at http://tinyurl.com/mwqspgy.*

—Lisa Kaczmarczyk

Mark Guzdial is a professor at the Georgia Institute of Technology.

© 2015 ACM 0001-0782/15/06 $15.00

Science | DOI:10.1145/2754949 Neil Savage

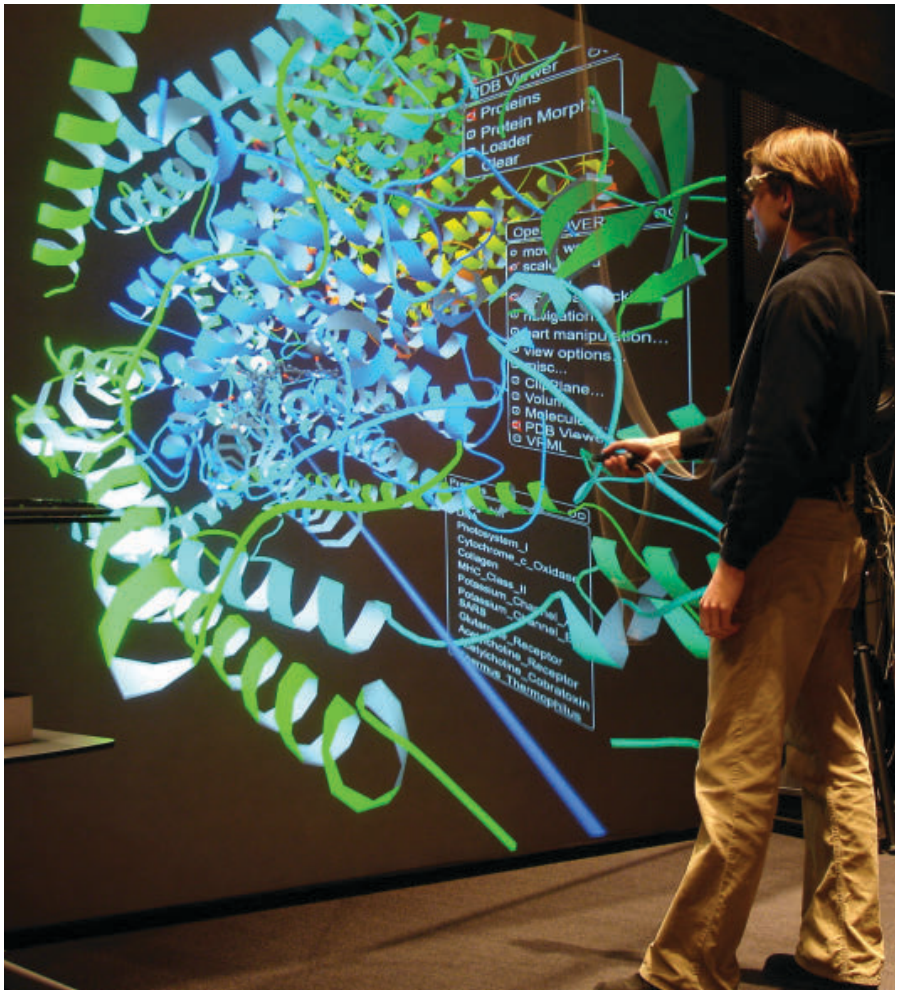
Plenty of Proteins

The growth of structural biology brings new challenges for the world's protein data archive.

In biology, structure is important. The proteins that build our bones and skin, digest our food, replicate our DNA, and perform all manner of other tasks are built from amino acids, and fold themselves into shapes that dictate their behavior. For decades, structural biologists have been measuring and collecting protein structures and depositing them and their chemical formulas in a computer archive known at the Protein Data Bank.

What began in 1971 as a group of seven protein structures has ballooned to a collection of more than 107,000 biological macromolecule structures, with about 10,000 added annually. Many are much larger and contain much more information that the original seven, which creates new challenges for the computer scientists in charge of maintaining the archive. "Because the structures are so complex and the information is rich, the bigger structures now are so big that we had to move to a new format for archiving them," says Sameer Velankar, team leader of the content and integration group of the Protein Data Bank in Europe.

The PDB, as it is known, consists of four data centers; at the European Molecular Biology Laboratory in Hinx-



An interactive visualization of proteins from the Protein Data Bank on a virtual reality wall in the California Institute for Telecommunications and Information Technology at the University of California, San Diego.

IMAGE BY JURGEN SCHULZE, UC-SAN DIEGO. COURTESY OF NATIONAL SCIENCE FOUNDATION

ton, UK; Osaka University in Japan; Rutgers University in New Jersey, and the supercomputing center at the University of California, San Diego. Those may be joined, in a few years' time, by partners in China, India, and Brazil. The PDB provides data to all users free of charge, and has become such a central resource that many scientific journals require protein models to be deposited in the PDB as a condition of publication. Pharmaceutical companies often turn to the archive when they are searching for molecular targets for potential new drugs.

Part of the challenge grows out of innovations in the technology used for studying protein structures. The standard technique for determining protein structure, dating back to the 1930s and still accounting for more than 90% of the structures in the data bank, is X-ray crystallography. Scientists crystallize a protein in which they are interested, and then scan it with an X-ray beam; the planes and edges of the crystalline faces scatter the X-rays, producing diffraction patterns that provide a precise description of the structure of the protein.

Not every protein, however, lends itself to easy crystallization. Biologists are also studying larger, more dynamic structures—"molecular machines" such as the ribosome, the factory within each cell that reads the RNA and uses that information to build new proteins. Ribosomes and other machines contain a variety of individual proteins and change their behavior, and thus their structure, over time. "It's hard to grow crystals from mixtures of different states," explains Stephen Burley, a professor of chemistry and chemical biology at Rutgers and director of the PDB data center there. "We're tackling machines that are bigger and bigger, more and more complicated, with more and more moving parts."

Biologists have turned to other imaging techniques. Nuclear magnetic resonance (NMR), for instance, is nondestructive, and allows scientists to watch molecular machines in action. Cryo-electron microscopy requires freezing samples at liquid nitrogen temperatures (−346°F to −320.44°F), but keeps them in a more natural condition than crystallization does. The PDB only contains experimentally derived structures. For computer-generated structures, biologists can turn to the National Institute of Health's Protein Model Portal. Neither does the PDB address folding dynamics, about which not much is known, Burley says.

A team of biocurators examines the data as it comes in, validating and annotating it. They represent on a color bar how well a model scores on a number of criteria, so a user knows how much confidence to place in a model.

With this variety of new techniques, the archive can provide a richer variety of information, but not always the same information from one model to the next. "All give little bits and pieces of the puzzle, but not with the level of detail that the traditional methods do," says Gerard Kleywegt, head of the European PDB. "For these new models with these techniques that have relatively low information content, maybe all you can track is the rough shape of the whole molecule."

That puts the burden on the archive to make it clear just how the models were developed, what assumptions went into making them, and what research questions they can or cannot answer, so they can correlate data derived through different methods. "We have to make it clear to our users what the structures are good for," Kleywegt says. To achieve that, the PDB has developed standards for validating the models, measuring how well they fit with experimental data so that fit can be expressed in numbers. A team of biocurators examines the data as it comes in, validating and annotating it to make it usable by biologists. They

ACM Member News

HEARST STRIVES TO IMPROVE CS EDUCATION



Marti Hearst thrives on utilizing cutting-edge computing technology to ameliorate education. A professor in the University of California Berkeley School of Information who also holds an affiliate position in the school's Computer Science Division, Hearst's primary research interests include search engines and their user interfaces (UIs), information visualization, natural language processing, and improving the massive open online course (MOOC) experience.

"There's lots we can do with computers in education. We're laying the groundwork for an aspect of visualization that's exploratory data analysis—an ill-defined field until now, and finding ways to improve MOOCs," Hearst says.

Hearst received bachelor's, master's, and doctorate degrees in computer science from the University of California at Berkeley. After graduation, she spent three years on the research staff of Xerox PARC, a stint she credits as "hugely influential and critical" to her current visualization, UI, faceted navigation, and computational linguistics work. "I could not have done what I did if not for PARC," she says.

At Berkeley since 1997, Hearst stays at the forefront of technology. The MOOC Chat Project, which she launched last October, aims to overcome the isolation of online learning via active peer learning. The project puts students into small groups; the instructor pauses every few minutes for active discussions to foster better comprehension and improve grade point averages. "I find it exciting and inspiring that the MOOC Chat Project can help many different students," Hearst says.

Hearst hopes to construct an Essay tutor package; "starting out with something that doesn't work well and based on the mistakes students make, we'd refine it iteratively," she says.

—Laura DiDio

represent on a color bar how well a given model scores on a number of criteria—redder for a low score, bluer for a better one; that way, a user knows how much confidence to place in a model, even without a deep understanding of the criteria that went into the scoring.

One approach to combining data derived from different techniques is software called the Integrated Modeling Platform (IMP), developed by Andrej Sali, a professor of bioinformatics and computational structural biology at the University of California, San Francisco. He and his colleagues created open access software designed to be flexible and general enough to build models using data from the various imaging methods, and to allow scientists to share and add to each other's models. Traditional modeling programs, Sali says, have lacked the necessary flexibility.

IMP treats a model as a collection of particles, with each particle representing some piece of the system. If the researcher has data on the arrangement of the atoms in a crystal, each particle could represent an atom, producing a high-resolution model. With lower-resolution data—say, measurements of distance between different components of a protein—the particles could represent those components. Different parts of the model might have different resolutions depending on the source of the data. Each particle has attributes associated with it, such as its radius, mass, or location.

The software uses this information to create a series of models that could fit the data. It then applies a scoring function to check the models, looking to see how well they match with information known about the particular structure being studied, as well as with structures in general. "You may find some problematic data that way," Sali says. "You can then go back and maybe collect some more data, or check the accuracy of the data."

Sali's hope is that researchers contributing to the PDB can easily look at other groups' models and check them against their own data, as well as speed up their own work by relying on models developed by others. "Why should anyone have to reinvent the wheel for any part of a bigger problem they're trying to solve?" he asks. "They should just be able to borrow it."

Sharing models between research groups and computing with data derived from multiple sources are issues faced by a growing collection of databases, says Manish Parashar, a professor of computer science and head of the Rutgers Discovery Informatics Institute. The Institute runs large-scale data analytics and simulations of data from a wide variety of sources, including the PDB. For instance, it models the growth of tumors and how they react to drugs for the Cancer Institute of New Jersey, combining data from sequencing the tumor's genome with NMR or (Magnetic Resonance Imaging (MRI) images.

The curators of any set of data have to figure out what metadata to associate with it to make it useful and searchable by the people who will base their research on it, asking, "What makes sense at the science level? How do you encode that into an algorithm?" Parashar says. For example, if the researchers will be asking questions about the shape of a protein, the data has to be labeled with commonly accepted terms describing shape so the researcher can run queries on those terms.

Scientists operating the PDB continue to work on ways to make data shareable and useable. Last October, they held a meeting in Hinxton to discuss how to deal with hybrid models and the expected growth of their archive. They are optimistic they can meet the challenge. "It was started early enough that we can be ready when these big structures start coming in," Velankar says.

The scientists are aware of the problems that can arise when data standards become outmoded; PDB files were originally written in Fortran, created on punch cards, and sent through the mail. The file structure was based on the cards' 80-column format and could not accommodate structures with more than 99,999 atoms. As larger structures were discovered, those files had to be broken into separate chunks. "The ribosomes were so big that they actually broke the old data format system," Burley says. The archive has since changed to a new format, and updated its software to work with it.

One non-scientific challenge is the issue of how to pay for an archive that is expected to keep growing, and that will require more storage space and more biocurators to process the incoming data. The government grants that fund the project have been flat for the past several years. "It's a continuous struggle to try and find money," Kleywegt says. Storage is not much of an issue; the archive contains approximately 546GB of data. Downloading the data, especially the full 3D images of large macromolecular machines, can cause "significant capacity issues," Burley says. On average, users download a million individual structures every day, which means the archive must deliver approximately 7TB each month.

"We're projecting five years from now the archive will be at least 50% larger and the complexity will be considerably higher," Burley says.

Despite the challenges, Burley finds the accelerating pace of structural biology and the adoption of new data collection methods that are driving the growth very exciting. "It's a good problem to have," he says, "The science is moving very fast." C

Further Reading

Russel, D., Lasker, K., Webb, B., Velazquez-Muriel, J., Tjioe, E., Schneidman-Duhovny, D., Peterson, B., and Sali, A.
Putting the Pieces Together: Integrative Modeling Platform Software for Structure Determination of Macromolecular Assemblies, *PLOS Biology 10*, 2012.

Berman, H.M., Kleywegt, G.J., Nakamura, H., and Markley, J.L.
The Protein Data Bank archive as an open data resource, *J Comput Aided Mol Des* 12, 2014

Cossio, P., Trovato, A., Pietrucci, F., Seno, F., Maritan, A., and Laio, A.
Exploring the Universe of Protein Structures beyond the Protein Data Bank, *PLOS Computational Biology 6*, 2010.

Bourne, P.E, Westbrook, J., and Berman, H.M.
The Protein Data Bank and lessons in data management, *Briefings in Bioinformatics 5*, 2004.

RCSB Protein Data Bank – Highly Curated Data https://www.youtube.com/watch?v=arO8EgsKvo8

Neil Savage is a science and technology writer based in Lowell, MA.

© 2015 ACM 0001-0782/15/06 $15.00

Technology | DOI:10.1145/2754954 Samuel Greengard

Between the Lines

Smartphone apps are driving changes in the way people park. Sensors, crowdsourcing, and big data are making it easier to find open parking spots.

It is no secret that one of life's thorniest and most frustrating challenges revolves around finding a parking spot. Every day, in a scenario that plays out in urban areas around the world, motorists cruise, circle, and dart through streets in search of an elusive place to stow their vehicle for a few minutes or a few hours. Too often, parking lots are full and street spaces are next to impossible to find. The process is incredibly inefficient and time-consuming—and it frays nerves and burns gasoline.

University of California, Los Angeles (UCLA) distinguished urban planning professor Donald Shoup has estimated motorists searching for street-side parking spots in Los Angeles accrue 950,000 excess vehicle miles per 15-block area over the course of a year. This extra driving—which accounts for about 30% of traffic in business districts—consumes approximately 47,000 gallons of gas and generates about 730 tons of carbon dioxide annually. However, those factors are only parts of the problem. Many people overpay for metered parking in order to avoid costly fines. "The simplest, cheapest, and fairest way to reduce traffic congestion is to set the right prices for on-street parking," Shoup says.



SpotHero

The biggest problem with parking apps, for now, is simply attracting enough users to make the apps valuable.

"It's an incredibly inefficient system," says Geoffrey Challen, an assistant professor at the State University of New York at Buffalo (SUNY Buffalo).

However, all of this may soon change. Smartphones, apps, big data, and increasingly sophisticated algorithms are introducing new ways to address everything from the probability of finding a parking spot at any given moment to pocketsourcing methods that gauge the actions and behavior of motorists. Although researchers and private companies are approaching the task in fundamentally different ways, the common denominator is that it may soon be a lot easier to hop in a car, drive to a destination, and find an available parking spot there.

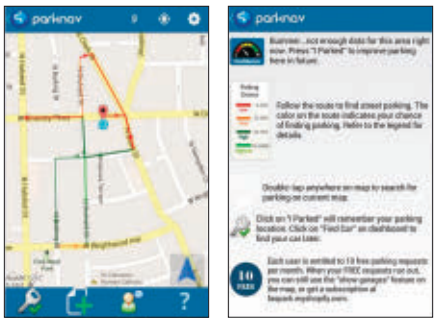
"Mobile technology and sensors are fundamentally changing the equation," explains Eyal Amir, a professor at the University of Illinois at Urbana-Champaign and CEO of Incube, Inc., a firm that collects street-level information and uses it to recommend driving routes and parking options through an app called Parknav. "With the availability of big data, we are able to understand and address parking in a fundamentally different and better way. We are able to introduce efficiencies that weren't possible in the past."

On the Spot

As cities and individuals look for ways to reduce congestion, save fuel and time, and improve access to shops and other businesses, parking is clearly in the crosshairs. For example, in Portland, OR, sensors in parking structures at Portland International Airport detect whether vehicles occupy spots. A red light above a space indicates it is taken, and a green light suggests the spot is available. At the end of each row, signs display available spaces. These same systems have been used in Europe and other locations for years.

In other instances, lots and apps now book reservations or auction off parking spots, though some cities, including Los Angeles and San Francisco, have attempted to shut down these services, particularly when they involve private parties holding and auctioning off public spots and spaces. Some app operators have attempted to pay individuals to grab spots on the street, hold them and then auction them off, often for $4 to $5.

Trying to make sense of parking and understand where and when spots might be available is no simple thing. The task is typically based on a combination of three elements: sensing or crowdsourcing, understanding human behavior, and accumulating massive data that is constantly undergoing change based on both



ParkNav

predictable (holidays) and unpredictable (weather) events. Says Cecilia Mascolo, professor of mobile systems at the University of Cambridge in the U.K., "This is all about how you can use sensors to better understand human behavior. People respond to parking challenges in very personal and specific ways, but there are overall patterns."

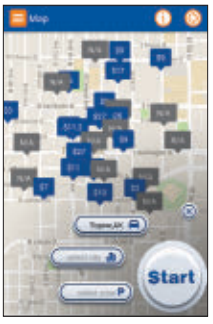
Amir says the introduction of the iPhone, and then Android smartphones, changed the stakes. Most importantly, they introduced a platform for connecting people into a broad human network and using common apps and interfaces to collect, manage, and apply data. The ability to carry them in pocket or purse—and create a persistent connection—was revolutionary. Meanwhile, smartphones have advanced radically over the last eight years. Sensors and increasingly powerful processors—along with crowdsourcing, big data, cloud computing, and artificial intelligence—have made it possible to detect and decipher parking patterns in ways that once would have been unimaginable.

Over the last few years, a spate of parking services and apps has appeared.

Some, such as Pango Mobile Parking, allow users to pay for street parking or spaces at private lots using their smartphones. The service is live in more than 60 cities worldwide.

Others services, such as ParkingSpotter, BestParking, Parkopedia, SpotHero, and ParkMe, help users find and reserve spots through a website or mobile app. Typically, these services display a map with pins displaying current prices. In many cases, these apps require cooperation with parking facilities or government agencies. They often rely on magnetic loops and Wi-Fi to count vehicles and transmit data.





Pango

While these services may make it easier for some people to park some of the time, they do little to address overall load demand and the inevitable parking crush that hits cities during business hours and at other peak times. They also do not help individuals find specific spots. This has led researchers, including Challen, Amir, and Mascolo to dig deeper into the parking challenge and look for ways to reduce the need to circle blocks or structures searching for a coveted spot. Yet even with sophisticated sensors, algorithms, and APIs to move data between systems, this is an incredibly challenging task.




Parkopedia

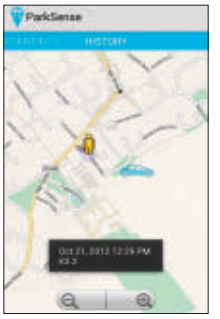
Forward Thinking

Most researchers and app developers aim to make it easier to find a parking spot while avoiding expensive technology and infrastructure upgrades.

Challen and a team at SUNY Buffalo have explored the concept of pocketsourcing by developing an Android-based app. It relies on GPS and accelerometer data from smartphones to record locations and movements of participants. There are no technology investments or upgrades for lots, and users do not have to click any buttons. The app sends data to a server, which plugs in data from OpenStreetMap; the aggregated data reveals the odds that a space is available at a particular location.

The system makes inferences about user activity based on specific movement characteristics. It can differentiate walking from the motion of a vehicle and even can identify different speed and movement patterns for cars. Challen's team tapped 105 smartphones around Buffalo, NY, and, over a 45-day span, studied more than 10,800 arrivals and departures using a camera to verify the accuracy of the algorithm. They found they could predict the availability of parking spots at about a 94% accuracy rate.

At the University of Cambridge, Mascolo has focused her research on street parking. "People have trouble finding places to park, they overpay, and they wind up getting fined," she says. The team developed a smartphone-based sensing system that taps into accelerometers and gyroscopes on the phone as well as surrounding Wi-Fi beacons to sense when people leave parking spots. Using a Wi-Fi signature-matching approach that compares the phone and the beacons, it is possible to detect when a driver returns to a parked vehicle and begins driving. "We know the Wi-Fi stations around the vehicle and that creates a specific fingerprint of the place," she says. The system, called ParkSense, not only identifies when a spot becomes available, it introduces a pricing model based on actual usage. She studied 484 drivers and found the system reduced driving time by 15%.





ParkSense

While Challen and Mascolo have studied parking and conducted academic research on the topic, Amir has started a commercial company named AI Incube Inc. and developed an app that aids in finding parking spots. The firm's Parknav app works in Chicago, San Francisco, Munich, and Hanover. It continually collects data about motorists as they move in and out of parking spots and constantly updates the underlying algorithm through machine learning, rather than traditional statistical methods. This not only helps refine the algorithm dynamically as data flows in, but it also detects variations based on the day of the week, holidays, or when sporting events take place.

Gigabytes of data per day flow into the system, which adjusts dynamically



ParkMe

using real-time GPS data along with proprietary and public data sources in order to become more accurate over time. The company claims the system reduces street parking time by 70% or more. ParkNav, which offers iPhone and Android apps for free (it currently boasts about 120,000 users), makes money by selling data to the automotive industry, companies that operate fleets of vehicles, and large real estate firms looking to optimize vehicle fleets. The data consumers generate when they use the app might also benefit governments, particularly cities, Amir says. "It has the potential to redefine the way we approach parking and introduce far more efficient systems."

Driving Progress

The biggest problem with parking apps, for now, is simply attracting enough users to make the apps valuable. Almost all plug-in data comes from a relatively small percentage of motorists. As Amir puts it, "If you have 95% of the population using an app, you can generate extremely accurate data. If you have only 33% using the app, you have data from one [vehicle] but you are missing the other two [vehicles]. As a result, you have to play more games with the data. You have to use more hints—about the desirability of a parking spot or popularity of a particular lot—and make more projections." A problem with making assumptions, Amir says, is that people do not behave consistently or in a predictable way.

In addition, even the best sensors and algorithms can be fooled. For example, a passenger in a vehicle may access the same app the driver is using; in that instance, the system may detect two cars pulling away from a spot rather than one. That, in turn, introduces another problem. "Crowdsourcing and pocketsourcing are extremely powerful tools," Challen says, "but the more you ask users to do, the less they participate and the more prone they are to forgetting or ignoring requests for data input. They may also be a couple hundred feet away or it may be a few minutes later, and a parking spot is already taken."

Nevertheless, parking apps are rolling forward and growing in appeal. Researchers say that within the next decade, they will play a crucial role in finding parking spots and reducing urban congestion. They may also introduce fairer parking pricing models that are based on actual time used rather than pre-paid blocks of time. As Shoup puts it, "The recent flowering of technology to measure on-street parking occupancy and to charge variable prices for on-street parking now allows cities to set these right prices."

Says Amir: "The key is to get people to understand the value of the technology and get major companies like Google to embed it into mapping apps and other tools so that it can be used in a more intuitive and useful way." C

Further Reading

Nawaz, S., Efstratiou, C., and Mascolo, C. ParkSense: A Smartphone Based Sensing System For On-Street Parking. *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*, MobiCom 2013, ACM, pp. 75–86. http://bit.ly/1zPXu0M.

Nandugudi, A., Ki, T., Nuessle, C., and Challen, G. PocketParker: Pocketsourcing Parking Lot Availability. State University of New York at Buffalo. http://bit.ly/1DYSLdI

Propst, J., Poole, K., and Hallstrom, J. An Embedded Sensing Approach to Monitoring Parking Lot Occupancy. *Proceedings of the 50th Annual Southeast Regional Conference*, ACMSE 2012, ACM, pp. 309–314. http://bit.ly/1y8h9Mb

Han, J., Owusu, E., Nguyen, L., Perrig, A., and Zhang, J. ACComplice: Location inference using accelerometers on smartphones. In *Communication Systems and Networks* (COMSNETS), 2012 Fourth International Conference on, pp. 1–9, January 2012.

Samuel Greengard is an author and journalist based in West Linn, OR.

© 2015 ACM 0001-0782/15/06 $15.00

Milestones

HiPEAC Unveils 5th Vision Document

The European Network of Excellence on High Performance and Embedded Architecture and Compilation known as HiPEAC (High Performance and Embedded Architecture and Compilation) recently released its fifth HiPEAC Vision document (http://bit.ly/1wyXYu5) presenting recommendations for Horizon 2020, the European Union's 80-billion-euro Research and Innovation program for 2014–2020.

The mission of the HiPEAC network, which includes "about 1,500 researchers in computing systems in Europe," according to the organization's website, www.hipeac.org, is "to steer and increase the European research in the area of high-performance and embedded computing systems, and stimulate cooperation between academia and industry and computer architects and tool builders."

The network is run by a consortium of six universities, one research institute, and five companies, and is coordinated by Belgium's Ghent University.

The organization says recommendations in its latest Vision document are "based on an analysis of market trends, a discussion of technology constraints and opportunities, and a review of Europe's strengths and weaknesses in the field of computing systems."

Marc Duranton, chief editor of the Vision document, said its key theme "is the imminent end of uninterrupted exponential growth in computation, communication, and storage capacity with potentially disruptive consequences for the global economy."

HiPEAC network coordinator Koen De Bosschere, a professor at Ghent University, called the Vision document "one of the key documents steering the research funding in computing systems in Europe.

"We believe it is definitely important for the ACM Europe community, but also relevant for the global ACM community."

—Lawrence M. Fisher

Society | DOI:10.1145/2754951 Gary Anthes

Estonia: A Model for e-Government

Over the next decade, the population of Estonia is expected to soar more than 600% as the country becomes the first in the world to open its borders to an influx of e-residents.

OVER THE NEXT decade, the population of Estonia is expected to increase more than 600%, from 1.3 million to 10 million.

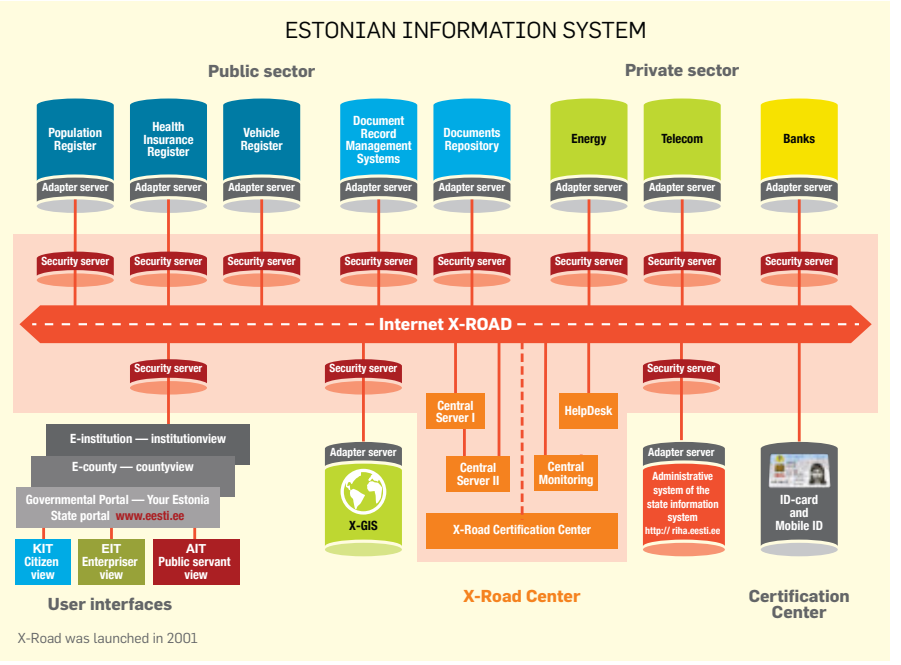
No, a flood of immigrants is not likely to invade the tiny Baltic republic. This year, Estonia becomes the first country in the world to open its borders to e-residents—people who sign up for digital identification cards and digital signatures for access to a wide array of national electronic services and databases. The idea is to provide a gateway by which people outside of Estonia can make investments in Estonia, establish businesses there and, eventually, use the country as a bridge to commerce elsewhere in the European Union.

The bold plan is a logical step forward in an unprecedented roll-out of e-government services that began in 2000, when Estonia introduced a public system for electronic tax filing. In 2002, Estonia introduced a universal electronic identification card with digital signatures, which every citizen gets at the age of 15. The ID cards and signatures have become the keys to nearly universal access to government information and services as well as private-sector services in health care, banking and education, and law. In the years since, the Estonian government and industry have put more and more functions online, all connected by a nationwide data backbone called X-Road.

Digital signatures in Estonia are so widely used and trusted that they are now preferred to signatures on paper, says Siret Schutting, managing director of e-Estonia Showroom. She estimates digital signatures save the country 2% of GDP, or $500 million a year. "We can use it everywhere," she says. "It's legally the same as a handwritten signature."

The Estonian e-tax system, along with X-Road's access to multiple government and private sector databases, has made tax filing unimaginably simple, at least by U.S. standards. "No one even considers paper as a possibility anymore," says Siim Tuisik, an Estonian citizen lobbyist for open data and digital rights. "If you are a private citizen, you make five or six clicks. All the data is already there; everything is prefilled, so unless something is wrong, you don't need to add anything."

Other countries, especially in Europe, are eyeing Estonia's successes with envy. "Estonia is way ahead of other countries, really a model," says Clare Sullivan, a cyber-lawyer and professor at the University of South Australia School of Law. "They are where we are all going to be in five to 10 years." Estonia has the most advanced e-society in the world, with the government offering some 600 e-services to citizens and 2,400 to businesses, she says.



The databases and other components that make up X-Road, the backbone of e-Estonia.

Why Estonia?

Sullivan says Estonia's progress is due in part to a strong commitment to Information Technology (IT) on the part of the prime minister and other senior government officials, stemming from the country's independence from the Soviet Union in 1991. "It was very poor, and struggling to establish its own identity and own economy," she says. "They had few resources otherwise, so they decided to go with IT."

"Information technology allows us to achieve beyond our natural means as a small government and country," said Prime Minister Taavi Rõivas in an interview. "That is why we have implemented a trusted national identity scheme, universal data exchange layer X-Road, various digital services and initiatives in all walks of life." He said he is deeply involved in the roll-out of Estonia's digital services as chairman of the E-Estonia Council, which "steers

INFOGRAPHIC COURTESY OF E-ESTONIA.COM

the making and execution of national digital agenda in the country."

Cultural and environmental factors have also driven Estonia's remarkable progress into digital services. The public puts a great deal of faith in the integrity of the systems and databases linked by X-Road. "The idea of people not trusting government is, if I may say so, a very American thing," Sullivan says. "In Europe, distrust is centered more on business than on government, and in Estonia, it's become part of the culture. They have been doing this since the early 2000s, and they are hooked on the convenience of it."

Estonia is smaller than West Virginia, and the third-poorest of the 28 countries in the EU. Not surprisingly, a number of high-tech start-ups in the country have focused on cross-border flows. For example, Skype for voice and video communications, and TransferWise for moving money, have their origins in Estonia.

"Nordic countries strongly believe in the inclusion of different social groups, and the digital solutions help do that," Schutting says. "Access here is not a privilege, it's a right." Indeed, she says, even the poorest citizens have free Internet access at libraries, banks, post offices, and other places. A project begun in 2009, called EstWin, aims to bring 100 Mbit/s Internet access to every citizen of Estonia by the end of this year. By 2018, the speed for many users is due to be boosted to 2.5 Gbit/s.

Elsewhere in the World

Estonia may be a world leader, but it is hardly alone in its moves to public e-services. Last year, Estonia joined with the United Kingdom, South Korea, Israel, and New Zealand to establish an initiative called D5. At the first annual D5 Summit in London last December, the five countries drafted a charter pledging to work together and share ideas to develop and integrate electronic public services.

D5 emerges from the wreckage left by the U.K.'s failed rollout of a national identity card system several years ago. The huge, and hugely expensive, card project, along with a proposed National Identity Register, were scrapped in 2010 in the face of strong opposition from the public and from some quarters in government over concerns about an erosion of civil liberties and privacy. Estonia was made a charter member of the D5 by other members looking for guidance on these kinds of efforts, according to citizen activist Tuisik. "We get about 400 official visits each year that concentrate on digital matters," he says.

A big challenge in developing governmental e-services is keeping them decentralized for security, while preserving cross-application compatibility.

In a recent blog, Sullivan called D5 an important development because it is an international acknowledgment of the importance of e-government. "[It] formally acknowledges the need for international collaboration to ... develop common standards and procedures," she says.

However, notes Sullivan, although the D5 calls itself "a group of the world's most digitally advanced governments," its membership roster is conspicuously missing Singapore, India, Australia, other European countries, Canada, and the U.S. "Wider membership could facilitate greater exchange of experience and cooperation as well as development and implementation of international standards," she says.

Separately, Finland is marching to Estonia's drum and is now building an electronic identification system and a national data exchange layer for electronic services based on Estonia's X-road concepts.

India has embarked on a massive project to collect biometric data—including a photograph, 10 fingerprints, and two iris scans—from each of its 1.3 billion residents. The government will use the data to build a giant "identity database," then use that in a companion project that assigns citizens 12-digit ID numbers and chip-based identity cards. The stated intent of the two projects is to better serve the public with a suite of government services and programs but, like the U.K.'s aborted project, the effort is controversial because some believe it will lead to violations of privacy rights.

Guidance

Tuisik says no effort has been spared to give users control over their own data. For example, patients can limit access to their electronic medical records to specific named individuals, and they can see at any time an audit trail of those who have accessed their records. "Doctors violating the rules will be sacked," he says.

A big challenge in developing governmental e-services is keeping them decentralized for security, while still preserving cross-application compatibility and data-sharing, Tuisik says. "Quite often in other countries, the main problem is that different parts of the government each offer their own services and gather their own data. There is no common backbone of data like X-Road, and no common ID numbers for users." Consequently, citizens see a government that is fragmented, rather than unified, he says.

Asked what advice he'd give a country like the U.S., Prime Minister Rõivas said, "I do not like to offer advice or teach others. Each country and government has different settings and context. However, I can reiterate and share the lessons we have learned, and it is up to U.S. readers and colleagues to see if they have relevance in that great country.

"We made a very conscious choice to build shared platforms, like joint digital identity and the whole-of-government data exchange layer X-Road, rather than allow diverse development. The idea was also to put in place the right conditions so that digital innovation can flourish in all parts of the public sector and society—bottom-up. If some area falls behind, we try to lead by example and kick-start things top-down, too. Despite such a rather decentralized approach, we make an

effort to ensure common standards and procedures are in place, so that in the end even bottom-up innovations are secure, interoperable, and convenient to use."

Security Risks

Nevertheless, government officials and users alike admit there are risks associated with X-Road and its attached databases and applications. "Everything is online," Schutting says. "What if something happened to our data? Our government wouldn't function anymore."

Something did happen, in 2007. A series of denial-of-service attacks brought down the websites of the Estonian parliament and other government agencies, banks, newspapers, and broadcasters. Estonia accused Russia of the attacks, and today global fears of state-sponsored cyber-warfare trace their origins in part to these network intrusions. Russia denied responsibility, and a year later an ethnic Russian man was fined for participating in the attacks.

Because X-Road and its applications form a highly decentralized system, the attackers could be isolated and then "snipped off" the X-Road before doing much damage, Schutting says. "But it was a wake-up call. It led to a real public discussion about data integrity and data protection." She says the attacks had a silver lining as they led to the creation of the NATO Cooperative Cyber Defence Centre of Excellence, based in the Estonian capital Tallinn, and to the *Tallinn Manual* on the application of international law to cyber-conflicts.

Despite these defensive moves, fears of another attack, including possibly a physical attack by Russia in the wake of its moves into Crimea and Ukraine, have prompted a plan by Estonia to put its e-government services "in the cloud," with its databases outside of Estonia in "data embassies" in friendly countries. However, that idea has prompted fears of breeches of data privacy by the law enforcement and intelligence agencies of those friendly countries.

In any case, the e-residency system for non-Estonians is moving ahead rapidly. It is in beta test mode now, and will go live sometime this year. At first non-residents will have to travel to Estonia to sign up, but sometime this year they will be able to do so at Estonian embassies anywhere in the world.

"This is literally pushing boundaries," Schutting says. "We are building a country without borders."

Further Reading

Aasmäe, Kalev
Estonia uncovered [blog], ZDNet
http://www.zdnet.com/blog/estonia/

Drysdale, Kirsten
Estonia offers e-residency to allow non-citizens access to government services and business online
ABC News Australia, Nov. 25, 2014
http://www.abc.net.au/news/2014-11-25/estonia-to-offer-e-residency-online-to-non-citizens/5917240 [Contains video]

Rõivas, Taavi
Leveraging technology in turbulent times (video), Stanford University Libraries, Dec. 10, 2014
https://www.youtube.com/watch?v=XOg0o3nmCYQ

Schmitt, Michael N. (Gen. ed.)
Tallinn manual on the international law applicable to cyber warfare, New York, Cambridge University Press, 2013
http://archive.org/stream/TallinnManual/TallinnManual_djvu.txt

Sullivan, Clare
Digital identity's new frontier: the political, economic, and legal ramifications of Estonian e-residency, Georgetown Journal of International Affairs, Nov. 27, 2014
http://journal.georgetown.edu/?s=estonian+e-residency

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

© 2015 ACM 0001-0782/15/06 $15.00

Milestones

Computer Science Awards, Appointments

ACM HONORS FIVE WITH TECHNICAL AWARDS

ACM recently announced the recipients of its 2014 technical awards.



Sylvia Ratnasamy, an assistant professor of computer science at the University of California, Berkeley, will receive the Grace Murray Hopper Award, presented annually to the outstanding young computer professional of the year, for "her seminal work on distributed hash tables, a fundamental contribution to the design of large-scale distributed and decentralized computing systems."



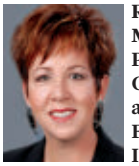
The Paris Kanellakis Theory and Practice Award honoring theoretical accomplishments that have had a significant effect on computing will be presented to James Demmel, a professor of computer science and mathematics at the University of California, Berkeley, for "contributions to algorithms and software for numerical linear algebra used in scientific computing and large-scale data analysis."



Jon Kleinberg, Tisch University Professor of Computer Science and Information Science at Cornell University, will receive the ACM-AAAI Allen Newell Award for his "groundbreaking work in computer science in areas including social and information networks, information retrieval, and data science, and for bridging computing, economics, and the social sciences."



William A. Wulf, professor emeritus of computer science at the University of Virginia, has been named to receive the Karl V. Karlstrom Outstanding Educator Award for his "contributions as a teacher, author, and national leader who focused attention and changed the national education agenda and in the process supported the needs of underserved and underrepresented students."



Robin Roberson Murphy, Raytheon Professor of Computer Science and Engineering Faculty Fellow for Innovation in High-Impact Learning Experiences at Texas A&M University, will receive the Eugene L. Lawler Award for Humanitarian Contributions within Computer Science and Informatics for her "pioneering work in humanitarian disaster response through search and rescue robotics, to the benefit of both survivors and responders.

The awards will be presented to the honorees at the ACM Awards Banquet in San Francisco this month.

In Memoriam | DOI:10.1145/2769955 Andreas Reuter

Klaus Tschira: 1940–2015



On March 31st, 2015, the entrepreneur, software pioneer, and patron and supporter of scientific research Klaus Tschira died unexpectedly at the age of 74.

He made numerous lasting contributions to the scientific community. Naming them all would exceed even a lengthy appraisal, so we will just mention the most prominent ones.

In 1972, Tschira and four colleagues started a software company that grew to become a world leader in business software: SAP AG. Today the company employs almost 75,000 people and has customers in 190 countries. Klaus Tschira was one of the key architects and developers for the core components of the SAP product suite. He used and, in many cases, developed ideas and concepts that in those days were just slowly gaining attention in the CS research community, concepts such as abstraction, modularization, adaptability, and extensibility. Unfortunately, he never published his ideas (other than by integrating them into the code base).

The fact that SAP software can easily adapt to changing legal and fiscal regulations—a big competitive advantage for the company—is largely due to the methodical developments in which Klaus Tschira played a key role.

In preparation of his retirement from SAP's Board of Directors, he established in 1995 the Klaus Tschira Stiftung, now one of Europe's largest privately endowed foundations. Its main goal is supporting research in the natural sciences, including mathematics and CS—always in combination with outreach activities to educate the general public about scientific methods.

To him, research funding was not limited to project funding; it included direct financial support for research organizations and for the necessary buildings. An example is the Advanced Training Center at the European Molecular Biology Lab, the internal structure of which is a double helix.

In 2010, he established the "Heidelberg Institute for Theoretical Studies (HITS)," a basic research organization covering subjects ranging from molecular biology to astrophysics, statistics, computational linguistics, and CS. Tschira was convinced that new breakthroughs will primarily be achieved at the interfaces between the disciplines. When deciding on a new research group, he followed the advice James Bryant Conant gave in 1945: "There is only one proved method of assisting the advancement of pure science—that of picking men of genius, backing them heavily, and leaving them to direct themselves." He supported his research groups generously, but he never tried to influence their agenda; it was the results he was interested in and the science in general. He was a regular at all the internal seminars, talking to the scientists at length—which occasionally resulted in the creation of a new research group.

Tschira was interested in an unusually wide range of topics, all of which he addressed with utmost meticulousness. His own role and importance he always tried to downplay. He kept in the background, avoiding public honors whenever possible. Nevertheless, he recognized the importance of the gifted individual for the progress of science, and tried to get such people interested in his ideas. This is the reason he enthusiastically embraced the idea of establishing the "Heidelberg Laureate Forum," where Turing Award recipients, Fields medalists, and winners of the Abel Prize meet with young scientists.

Klaus Tschira touched the lives of many and changed them for the better, be it in the context of SAP, in his foundation and research institutes, or through his many funding programs. His passing is a huge loss for the research community in Germany and beyond.

We will sorely miss the clear strategic thinker, the generous patron, the creative supporter of research, the lover of language who liked to play on words, and last but not least, the warmhearted, humorous, fatherly friend.

Andreas Reuter is managing director of the Heidelberg Institute for Theoretical Studies, scientific and managing director of the European Media Laboratory GmbH, and holds the endowed chair (supported by the Klaus Tschira Foundation) in Distributed Systems at the University of Heidelberg.

Copyright held by author.



Milestones | DOI:10.1145/2754956 Neil Savage

Forging Relationships

Michael Stonebraker didn't realize at the outset that it would take six years to create INGRES, one of the world's first relational databases.

In the early 1970s, a young assistant professor at the University of California, Berkeley, decided to build a relational database. The concept of such a database had been proposed in 1970 by Edgar Codd of IBM, who outlined the advantages it would have over the dominant database model at the time, IBM's hierarchical Information Management System.

Many computer science researchers had followed Codd's work with papers of their own, but none had gone beyond proof-of-concept prototypes. The Berkeley professor, Michael Stonebraker, wanted to build something that could actually work.

"This was new and different terrain, nothing like anything that had been done before," says Stonebraker, now an adjunct professor at the Massachusetts Institute of Technology (MIT) and recipient of the 2014 ACM A.M. Turing Award. Along with Berkeley computer science professor Eugene Wong, doctoral student Gerald Held, and some other students, Stonebraker spent the next six years building INGRES, one of the world's first two relational databases.

"Building a real database system is a huge amount of work," he says. "We didn't know how much work it was going to be. We just did it."

Hierarchical databases store information in a stacked set of categories. For instance, there might be a listing of departments within a business, and within those departments would be a list of employees. If, however, an employee did not belong to a specific department, or if the user did not know which department the employee was in, then a query would not be able to find information about that employee. Stonebraker felt users should not have to write an algorithm to find what they wanted, and their programs should not depend on exactly how the data was structured.

The result of their work was INGRES, for which Stonebraker, Held, and Wong were awarded the ACM Software System Award in 1988. The team made some choices that, as it turned out, helped boost the popularity of their system, says Held. They wrote it on an operating system that had recently been released by AT&T's Bell Labs, Unix, in a then-little-known language, C. As Unix spread, users found there was a ready-made database system they could use with it. Stonebraker and colleagues also made INGRES an open source project, which allowed users to refine and build on it.

As INGRES grew more popular, its limitations became clearer. It began to be used in areas beyond the business data processing it had been designed for, and it needed changes. That led Stonebraker to develop POSTGRES, which allowed users to add their own data types, from specific formats for dates and times to geographic points. This object-relational database was now able to work on a much broader class of problems.

By the early 2000s, the database world had changed, and businesses had mountains of data they wanted to query in various ways. That data was arranged in rows, and queries took a relatively long time to be processed, so Stonebraker introduced column storage, which he says was two orders of magnitude faster. These days, he is promoting the idea that a "one size fits all" approach to databases is one whose time is past. He has founded various companies—including Vertica, Streambase, VoltDB, and Tamr—to commercialize different software products he has developed.

Held, who was chairman of Vertica and is now chairman at Tamr—Stonebraker was chief technology officer at both companies—says it is unusual for researchers to straddle industry and academia for as long as Stonebraker has. "There aren't that many people that can have their feet both in the academic and commercial world," he says, adding that it may be that back-and-forth that has made him so productive. "Some people have one good idea and it's great and it's worth a lot. Mike has been in there designing year after year and decade after decade," Held says.

Stonebraker finds it important to go beyond simply describing ideas in a paper, to actually using those ideas to build practical applications. "If you don't have something people can use, you have no chance of making a difference," he argues. At the same time he was working on INGRES, IBM started developing its own relational database, System R. Stonebraker says the existence of each spurred the other along, and the two wound up providing valuable feedback to each other. "The way to do technological innovation is to build a better mousetrap and then threaten the existing order with your better mousetrap, and then something will happen."

Samuel Madden, a colleague at MIT who studies database systems, believes Stonebraker founded these companies to demonstrate that his academic ideas have real-world applications. "He's very singularly driven with a vision of the way the world ought to be," Madden says, "and remarkably most of the time this vision turns out to be right."

For undergraduates just getting into computer science, Stonebraker's advice is, "Learn how to code, and code well, because whatever you do is going to involve implementation." For Ph.D. students trying to figure out where to focus their attention, he suggests talking to real-world computer users. "They're happy to tell you why they don't like or do like any given technology, so they're a wonderful source of problems to work on."

Neil Savage is a science and technology writer based in Lowell, MA.

© 2015 ACM 0001-0782/15/06 $15.00

PHOTO BY JARED LEEDS PHOTOGRAPHY

DOI:10.1145/2753120

Vishal Misra

Inside Risks
Routing Money, Not Packets

Revisiting network neutrality.

There is an old conversation between two friends, a very prominent economist and Dave Clark (one of the foundational architects of the Internet) that is often recounted by Clark:[a]

Economist: "The Internet is about routing money. Routing packets is a side effect. You guys really screwed up with the money routing protocols."

Clark: "But we didn't design any money routing protocols!"

Economist: "That's what I said."

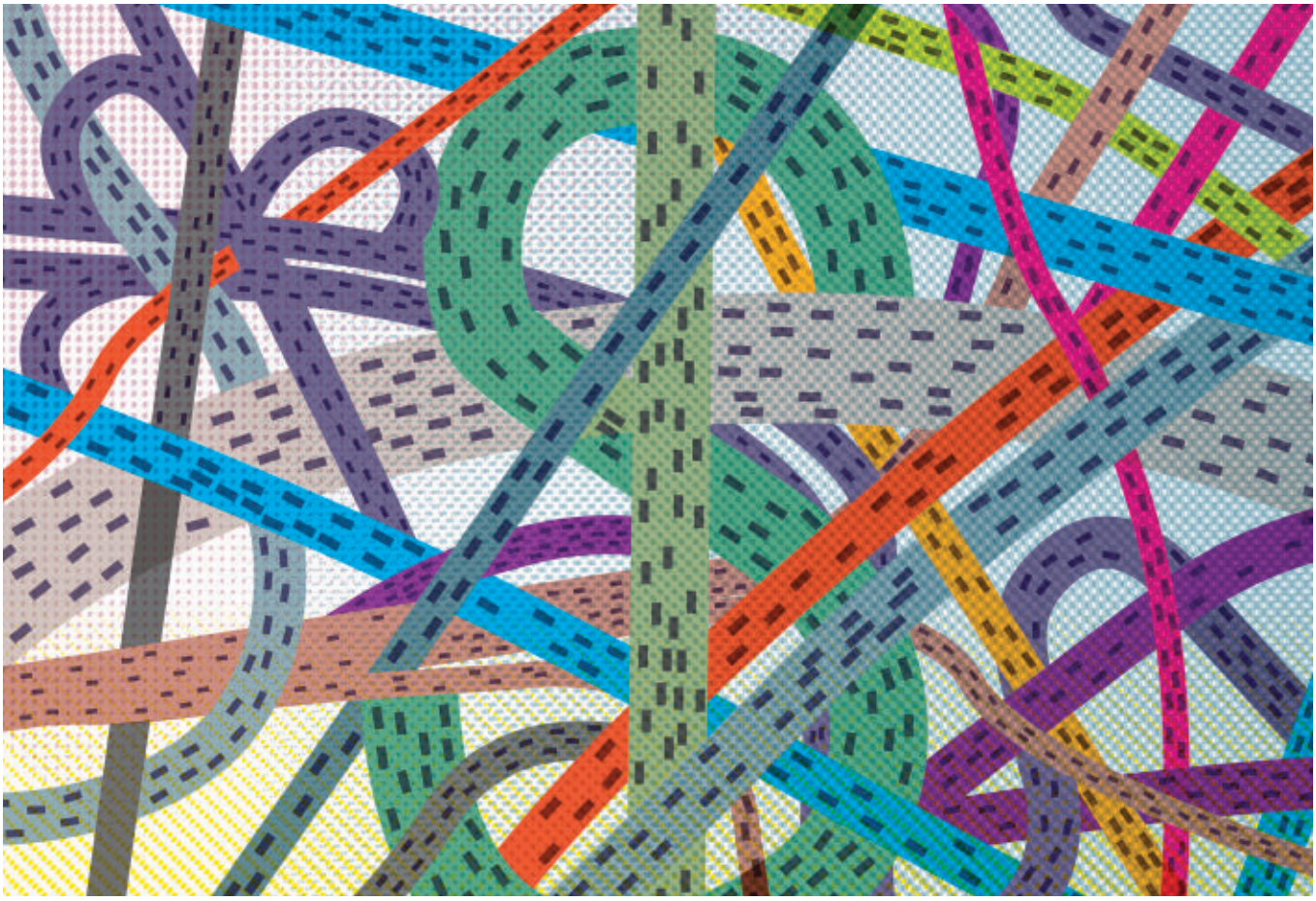
While this was lighthearted banter between friends, the economist had it right 15 years ago—the crux of the issue behind what is broadly termed as network neutrality was captured in the conversation. The Internet, which began its life as a network offering communication services between primarily non-profit entities such as academic and research institutions has now evolved into the backbone of unimaginable amounts of commerce. The raging debate over network neutrality can be explained by this changing nature of the Internet, the players involved, and the economic motivations. The FCC recently proposed Open Internet (the official FCC term for network neutrality) regulations, and invited public comments on it. Four million people responded, mostly in favor of the regulations. Finally, on February 26 the FCC had a vote (3-2) in favor of those regulations. While there was much celebration and joy among the activists on the success of the vote, there are some technical reasons we need to consider. In this column, I lay out those concerns.

While it is tempting to think of every packet being equal, the revenue each packet generates is different.

The Problems with the Ruling

The FCC proposed three "bright-line rules," namely "no blocking, no throttling, and no paid prioritization." One of the prominent reasons behind those rules is the very public dispute that Netflix had with major ISPs. However, there was no active blocking or throttling of Netflix traffic, and in the end Netflix quality has improved by a mechanism that amounts to paid prioritization but does not fall under the FCC's definition. The FCC has also said there will be "no unbundling," which is a mechanism that increases competition at the last mile. I make the case here that lack of competition at the last mile is really the core issue that needs to be addressed. The regulations also make no explicit mention of the practice of "Zero Rating" (where bandwidth consumed by certain content providers does not count against the users' quota). Zero Rating is a much more serious and real anti-

a https://www.amacad.org/content/publications/pubContent.aspx?d=1263



competitive problem than the ones the FCC has clamped down on (throttling/blocking/paid prioritization). In other words, what the FCC has missed is more important than what it is attempting to fix. Let me walk you through my argument for that claim.

The Changing Internet

Many of us remember the Internet from the days where all packets were indeed "equal" and the commodity that was traded among ISPs and between ISPs and customers was bandwidth. The unit of trade was normalized to bandwidth, and the applications on the Internet were primarily Web and email, and all packets could be treated uniformly and charged uniformly as well.

The online world looks remarkably different today, and the commerce the Internet generates goes beyond the trading of bandwidth. Trillions of dollars are generated using the Internet, whether we are talking established players such as Google and Netflix or upstart companies such as Uber or Airbnb. While it is tempting to think of every packet being equal, the revenue each packet generates is different—for example, for a search page (with ads) of Google versus a frame of a Netflix video. This difference needs to be modeled and accounted for.

This change in the economic value of traffic has thrown the economic stability of the Internet out of kilter. It shows up in disputes that are broadly classified as network neutrality issues. Before explaining the specific issues, I provide some background on Cooperative Game Theory.

A Cooperative Game Theory Lens

We looked at the Internet as an economic ecosystem, applying the lens of Cooperative Game Theory—specifically using the tool of Shapley Values. Cooperative Game Theory analyzes coalitions where a group of players (called a coalition) come together and generate some value, V. The solution of a cooperative game is the share of V that goes to individual players. A subset of cooperative games considers so-called convex games, where the marginal value of the coalition grows with the size of the coalition. Colloquially, this translates to "the whole is greater than the sum of the parts." The Internet is a good example of a convex game: various folk laws that define the value of the network as a function of the number of nodes as $V(n) = n^2$ (Metcalfe's law) or $V(n) = n \log n$ (Odlyzko's law) have shown good agreement with empirical data. Regardless of the precise form of the relationship, everyone agrees on the term "the network effect." There is general consensus on the "convexity" of the Internet.

Convex games have a set of solutions that are called the "core," where the share of the value that goes to each player creates an incentive for them to remain as part of the game. If there is a solution outside the core, some players have the incentive to leave the coalition, because they are better off not playing (either they are generating negative value because of costs involved, or they generate more value being on their own or part of some other coalition).

ILLUSTRATION BY MARK MCGINNIS

The Shapley value is a specific (and unique) allocation that can be axiomatically derived. One primary axiom that determines the Shapley value is that of "balanced contribution," which in simple terms means the value a player ends up obtaining is reflective of the contribution the player made. In convex games, this aspect of Shapley values shows up in the following way: the Shapley value solution lies in the center of gravity of the core; that is, it is the most stable of the solutions. It is the solution that keeps all the players as far away from leaving the coalition as possible.

The ISP Settlement Problem

We applied the Shapley value framework to analyze the complex topologies of the Internet and the different players such as ISPs, content providers, and (eyeball) customers. Following the taxonomy introduced by Dave Clark, we classified ISPs into three kinds:

- Eyeball ISPs that provide broadband connections to customers (such as Comcast and Verizon);
- Content ISPs that provide connectivity to content providers (we include content delivery networks in our definition), for example, Cogent, Level 3, Akamai; and
- Transit ISPs that provide global connectivity (such as AT&T, Telefonica, and Tata).

We modeled the revenue collected at the eyeball side as a flat-rate charge per customer (as is the norm for wired broadband), and at the content and transit level as a volume-based charge—which is again the norm. We also modeled the typical customer-provider relationships that exist on the Internet—with eyeball and content ISPs typically buying bandwidth from transit ISPs, and the transit ISPs themselves engaging in settlement-free peering with each other. As a thought experiment, we analyzed whether bilateral trading of bandwidth as a commodity could lead to a value allocation that was in the vicinity of the Shapley value, indicating a stable ecosystem. The idea behind our analysis was not to think of the Shapley value as a prescriptive solution for fair profit sharing, but rather look at the natural mechanisms that existed on the Internet and whether they end up in a stable region or not.

What we discovered through our analysis[1] was:

- As long as the flow of traffic on the network remained symmetric and the revenue collected at the two ends (eyeball end and content provider end) remained approximately similar, the bilateral trading in bandwidth resulted in a solution close to the Shapley value, indicating stability. This is reflective of the Internet of 15 years ago. Buying or selling of bandwidth resulted in a stable ecosystem.
- When the flow of traffic becomes asymmetric and the revenue generated at one end is imbalanced (on the content side, lump together the revenues that are being generated by Google, Netflix, and Amazon), the bilateral trading of bandwidth as the commodity leads to a solution outside the core. This led us to predict the rise of paid peering on the Internet to push the solution back inside the core. In simple terms, it is a way to transfer some of the value the content providers are generating (for example, Netflix subscription fees) to the ISPs to come up with a net value allocation that is in the core. The implicit trade that occurs now is not one of bandwidth, but that of revenue generated by content. Part of the revenue the customers of an eyeball ISP add to a content provider gets transferred to the ISPs via paid peering.
- The location of the solution also depended heavily on the level of competition. The more asymmetry in the level of competition, the less stable was the solution.

Our predictions started coming true in 2013–2014 when Netflix began signing paid peering arrangements with the major eyeball ISPs in the U.S. They ended up signing these paid peering deals with the top-four major ISPs in the U.S.: Comcast, Time Warner, Verizon, and AT&T. What is also interesting is all these four major ISPs are "local monopolies" in their respective geographical areas. The Netflix dispute with the ISPs has been at the center of the network neutrality debate, and the issue is purely economic. The public peering points between Netflix and the ISPs started experiencing congestion. However, this was not active throttling or blocking (practices the FCC is now explicitly prohibiting). The ISPs could afford to play hardball with Netflix because the customers of the ISP would not discontinue their broadband service if the quality of Netflix streaming was bad. The Internet provides enough other content for it to be worth it for the customer. In other words, there is a lot of "competition" on the content side of the Internet, but not enough at the eyeball side in the U.S., and that creates an asymmetry predicted and quantified by our analysis.

There is a lot of "competition" on the content side of the Internet, but not enough at the eyeball side in the U.S.

By signing these paid peering arrangements, the eyeball ISPs are able to monetize the excess bandwidth the content generates, and the value allocation solution returns to inside the core. What is also interesting is that while the network neutrality debate has centered on paid prioritization or throttling at the last mile, none of the ISPs were actually doing any of that. The congestion was happening at the peering links of Netflix (or their content delivery networks, Level 3) and the ISPs. By signing a paid peering arrangement, the ISPs have not created a "fast lane" but instead have created a fast "on-ramp" for the Netflix traffic that bypasses the open Internet peering points. The fight was economic, and the resolution was economic as well. There was no suppression of freedom involved—all the entities are interested in making money and that is the primary motivation. This problem arose because the level of competition at the last mile in the U.S. is extremely low, which throws the solution outside the theoretical core. The solution to this should be an approach that increases competition, such as unbundling or municipal broadband, rather than increasing regulation. A lot of places where there is thriving competition at the last mile (for example, Scandinavia and the

Far East) have no real or perceived network neutrality violations, and the Internet service there is fast and inexpensive.

Flat-Rate Versus Volume-Based Charging

A related issue is how ISPs charge for traffic. Customers much prefer paying a flat-rate amount, as it reduces their decision making to a once a month "is it worth it?" rather than saddling them with a constant flow of micro decisions for every connection to the Internet. For the ISPs it is bad news because they are unable to monetize excess traffic for flat-rate charging. In the wireless world on the other hand, the norm is volume-based charging (we include monthly quotas plus overage charges in the broad definition of volume-based charging). Unsurprisingly, debates about network neutrality do not appear there. It is in the interest of ISPs to have high-volume content such as YouTube or Netflix on their networks, because they can monetize that traffic.

The issue of Zero Rating is a related one that needs a careful examination here. In this scenario, content providers pay for the bandwidth the customers use for their content. This makes the ISPs happy as well, as they get paid for bandwidth—although this practice is truly anti-competitive. If a user gets to access Facebook or Netflix for free, then that user is unlikely to go to a different social network or streaming provider and pay for that bandwidth. For that reason, in the wireless world the big content providers are promoting Zero Rating—Facebook for instance is implementing something called internet.org in certain countries, where bandwidth for content from Facebook and other select sites is not charged to the customer. Zero rating is a much more serious threat to network neutrality than throttling or blocking. And the root cause is again economic. The content providers that seem to be on one side of the fight over network neutrality for the wired world are on the other side of it in the wireless world. Thus, it comes down to how the customer is charged—flat rate or per byte.

Innovation at Risk?

Network neutrality is an issue where the technical details get set aside for an idealistic statement that makes the case that "every packet should be treated equally." This is simply not true for modern networks—VoLTE is a very good example where voice traffic is prioritized over other traffic on LTE networks, bringing about the first significant improvement in voice quality in many years. According to the FCC ruling, if you are an application that is running with a public IP address, best effort is all you can get from the Internet. If any other kind of prioritized service is requested, it has to be classified as a managed or specialized service. This implies that a VoIP service from a cable provider can get prioritized service, but applications such as Skype and Vonage have to work with best-effort service. Similarly, health applications on smartphones or smart watches are becoming commonplace; under the new rules, an application such as dedicated heart-monitoring would have to come under the classification of specialized services, rather than use the data services already available on their smart devices. This has the potential to slow down innovation on the Internet, because the implication is that any kind of application that requires something more than best-effort service has a barrier to entry placed by regulations.

The network neutrality issue is really about economics rather than freedom or promoting/stifling innovation.

Differentiated services are routinely implemented in modern networks, and the designers of the Internet protocols foresaw this need by creating specialized Quality of Service bits as well as QoS architectures via a mechanism called DiffServ. The FCC ruling, however, can lead to the potential of the mechanisms remaining unexploited. There could be many other QoS sensitive applications that we cannot even imagine today that might need to run on the public Internet; we certainly have no idea where the world of Internet of Things would end up. It seems to be a technical risk to disallow any differentiation over the public Internet. The network should be allowed to differentiate but not discriminate, which is an important distinction that needs to be widely understood. Express delivery of packages in the physical world is allowed between all public addresses by all postal carriers, so there is certainly no reason to limit that capability on the Internet.

Conclusion

Our analysis over the past several years has revealed the network neutrality issue is really about economics rather than freedom or promoting/stifling innovation. This issue arose because the Internet has slowly transitioned from an entity that was primarily used for research purposes to the centerpiece of the modern economy. However, the reaction to the economic disputes that have naturally arisen has been to introduce regulations risking that they could potentially slow down if not completely limit innovation on the Internet. On the flip side, the lack of focus on the issue of competition can lead to a further strengthening of incumbent monopolies, which reduces the incentive for innovation. Making a policy decision without fully understanding and accounting for the science behind it is risky, from both an economics perspective and a networking one. The way forward lies in increasing competition at the last mile, rather than regulations that might impose technical limitations on the (future) functioning of networks. Network neutrality is the symptom; the real issue is lack of competition at the last mile. C

Reference

1. Ma, R.T.B., Chiu, D., Lui, J.C.S., Misra, V., and Rubenstein, D. On cooperative settlement between content, transit, and eyeball Internet service providers. In *Proceedings of the 2008 ACM Conference on Emerging Network Experiment and Technology* (CoNEXT 2008), Madrid, Spain, December, 2008.

Vishal Misra (misra@cs.columbia.edu) is a professor of computer science at Columbia University.

Copyright held by author.



DOI:10.1145/2753147

Peter J. Denning

The Profession of IT
Emergent Innovation

Fernando Flores, president of Chile's National Innovation Council for Competitiveness, discusses a new common sense about innovation.

In my March 2014 column, I commented on *Surfing Towards the Future*, a report from Chile that offers a new interpretation of how innovations arise.[1] The new story opens new possibilities for innovation in many areas including public policy, education, and world affairs. I recently discussed this with its main author, Fernando Flores.



Fernando Flores is an entrepreneur, philosopher, and engineer. In March 2010, Chilean President Sebastián Piñera designated him as President of Chile's National Innovation Council for Competitiveness.

Peter Denning: In *Surfing Towards the Future* you offered a new account of innovation. What drew you into that project?

Fernando Flores: Chile is a small country with the ambition to join the "Developed Countries Club." To realize its ambition, Chile needs before anything else its own understanding of innovation. Our report sought that understanding with a philosophical reflection on innovation. We wanted our pragmatic recommendations for public policy to be based on sound philosophy. We had intuitions that the current common sense about innovation is flawed. I think we did a good job telling a new story of how innovations emerge at particular moments in history when the conditions are ready. This story opens new actions for innovators and policy makers. We have been receiving very positive responses from around the world.

What do you mean by the flawed common sense?

The common sense is the shared presuppositions and sensibilities among people about what makes sense in their world. We call it a *sense* because it manifests as feelings and dispositions more than words.

The common sense about innovation is revealed in the popular stories that say that innovation is the product of creative people constantly inventing new technologies based in science. Yet innovations arise without creativity, new technology, or new science, and some creative technologies produce no innovation. Having a clear plan and strategy is also overrated. Some innovations come without these, and many plans and strategies do not produce innovation. *Innovation is a change of practice that displaces other practices already in place.* Inventing creative technology does nothing: people must take up new practice.

IMAGE COURTESY OF BIBLIOTECA DEL CONGRESO NACIONAL DE CHILE

A better story can be seen in Silicon Valley's practices dealing with global waves of change, which they detect, shape, and ride. We advocated that Chilean entrepreneurs visit Silicon Valley to transform themselves by immersing in its practices and thinking. They will surely make inroads when they return to Chile.

At the ACM 97 conference, you gave a speech about the impact of the Internet on business communications.[2] You predicted the importance of individual identities in the network of social connections that would emerge in the Internet. How is this connected with innovation?

The ACM 97 conference had a big challenge: to examine possibilities of the world in 50 years. I could not answer such a question by extrapolating trends visible in 1997. I noticed that computers had become significant in the world because of their powers of communication, not calculation. Since communication is a fundamental human concern, I speculated what might happen in a world where technology greatly enhanced communication.

I speculated that the network would be the primary medium for conversations and that people would develop network identities. Your identity would be the stories about how people perceive you and your ability to project into the world. I speculated that we would see an "age of identities" in the Internet, where our networked identities become of central importance to who we are. Amazon, eBay, Facebook, Twitter, and other social media services have realized this speculation. Identities are connected to innovation because they are sources of trust and social power.

I was not predicting the emergence of social media technology. I anticipated the emergence of a new space of possibilities. Social media have emerged in this space.

Why is the notion of historical emergence of a new space so important in your new story?

This is a very important question. The current common sense is blind to emergence. Emergence seems to be the essence of innovation. Our Western tradition interprets the world as a set of many objects. We innovate by making new objects from other objects. We are always using objects and moving among objects.

"I noticed that computers had become significant in the world because of their powers of communication, not calculation."

But this story misses something very important. The world is constituted not of objects but of practices. The practices are our automatic, embodied ways of thinking, feeling, and acting with each other. Objects are instruments that facilitate and enable our practices. We have individual, group, and community practices. We are not consciously aware of most of our practices.

Practices are deeply historical. The way the world works was not invented yesterday. It is a constant drift in our practices as we adapt to changes, circumstances, threats, and opportunities. Our current practices are the product of many years of historical evolution. Since innovation is change of practices, innovation is deeply connected to our histories. Our histories influence whether we are open or closed to various innovation possibilities.

Practices also confer power. People shape and effect change through the practices they already have in their networks and occasionally with new practices they invent. People who hold power will resist innovations if they are threatened by the loss of power from the new practices. The popular Uber ride-sharing company, for example, is meeting stiff resistance from taxi drivers and public officials in many cities. Because our common stories about innovation hardly mention power, many innovators are surprised when their proposals are met by indifference or outright resistance.

Because our practices in our worlds are always in flux, always drifting and changing, it is often impossible to assign any single agent to a change we see. Emergence describes our dawning awareness of the change that has already begun to gather momentum. The common notion that a creative idea triggered the change is usually wrong. The so-called creative idea is often a story we invent when we try to explain the emergence we are seeing. Historical emergence is part of a new common sense that sees the practices constituting the world. The emergence is contingent on the space of existing practices.

So, basically you are calling people to be much more sensitive to how the practices that sustain their worlds are already evolving and changing. The tension between possible future practices that are coming and our settled, habitual everyday practices is the cradle of innovation. The cradle is not in our mind or in our creativity. It is constantly emerging in our worlds. Can you say more why are you making room for *contingency* in your approach to innovation?

Innovations do not arise randomly, but depend on conditions and possibilities existing in the world. The World Wide Web gives a good example. Tim Berners-Lee invented a new technology in 1989 that combined technologies for markup languages, Internet naming, and hypertext. His original prototype ran only on NeXT computers. The new technology was contingent on the existence of the others. Berners-Lee was originally motivated by a concern to help the physicists in his lab (CERN) communicate better with their collaborators around the world. However, with help from his mentors at CERN, he advocated the Web as a new way for everyone to make connections and share information. That possibility intrigued a team at the University of Illinois, which built the first portable Web browser, Mosaic, an easily installed graphics interface to visualize the connection space of the Web. Entrepreneurs quickly began to experiment with businesses that would interact with their customers through a browser interface, a practice that gathered momentum and opened

the Internet to become a bustling space of commerce. Each of these elements—the original prototype browser, the combination of technologies, the development of a portable browser, and the commercial Internet—was contingent on particular policies, equipment, and practices at its time and each opened new spaces of possibilities on which subsequent innovations were contingent. Berners-Lee developed an uncanny ability to see the contingencies and mobilize people to follow them.

The important point is that innovations are contingent on the current world and the history of practices and technologies in the world. The world opens spaces of possibilities; people interplay and enrich those possibilities. The time becomes "ripe" for something new. But this dynamic process does not follow a linear causality to specific prior innovations.

Is this why many innovations seem to occur to unrelated people at about the same time?

Yes, I think so. *Innovation is always contingent on existing practices and concerns.* An innovator aims to take care of an unmet concern by proposing new practices that recombine existing practices. Often multiple members of the network will see an unmet concern and see ways to address it with existing and emerging technologies. Their proposals are often, to them, obvious responses to the unmet concern.

The ability to sense an unmet concern is very important. It is an ability to read moods and emotions by observing the conversations that people are having and the histories behind their practices. Often the first sense someone has about an unmet concern takes the form of an anomaly—the person is unsettled by something that seems out of place relative to our current practices and is not explained by our current practices. The person embarks on a quest to settle the anomaly and eventually designs proposals to help others with the same issue.

The phenomenon of multiple independent discoveries appears completely natural if you understand contingency and emergence. It also explains why it is so hard to find agents who caused particular innovations. Often there is no single agent and when you follow the chains of contingencies it is impossible to pinpoint a "cause" for an innovation.

Now I can see in your story why you are offering the metaphor of "surfing" to better grasp the phenomenon of innovation. The potential innovators are endlessly exploring, scouting, and diagnosing historical and emerging practices responding to a particular concern. That, I think, is why you speak about "immersing in" or "appropriating" traditions and practices. Can you speak more about your notion of appropriation and why it may be different from "information" or "know-how"?

My colleagues and I are still working on this story. *Appropriation* is a process of becoming familiar enough with a new domain to observe their practices, histories, and moods, carry on conversations with their people, see how power moves among them, and contribute to them. Often the work of appropriating a domain is motivated by the unsettlement we were just discussing; it helps to seek out interactions with other, unfamiliar domains to enrich your sensibilities and orientation. Through ongoing conversations, you will find those domains increasingly familiar and may discern practices that enable a response to the anomaly. You can then speculate about how to bring that practice to the original community. This first stage of appropriation might be called "advanced beginner" in the new domain and is sufficient to nurture new innovation proposals. As an appropriator, you are not a passive observer, but bring a self, an identity, a mood, and practices to the new domain, all of which affect the appropriation.

Appropriation is not the same as acquiring information or describing the skill sets of a domain, neither of which offer involvement in the practices of the domain.

We are looking for ways to help people appropriate unfamiliar worlds rapidly. Every domain has certain speakers who have an unusually good grasp and articulation of the central practices. If you can read, talk with, or watch such a person on a video, can you more quickly appropriate the domain. You do not have to agree with that person's views to let them help you appropriate their domain.

It is important to appropriate more than one domain because that expands the space of possibilities for innovations. In my own career, I have learned the domains of entrepreneurs, political leaders, business CEO, senator in a legislature, and more. I find that I am able to see bigger possibilities for innovation in any of these domains given that I can participate in the others and find connections. I have also found that watching videos is a powerful means of appropriating. YouTube and other online video services have provided means to learn new areas that simply did not exist a decade ago. The library called Conversations with History, founded by Harry Kreisler at Berkeley, is a magnificent example of a resource for appropriators—it features over 500 recorded interviews with history-makers in many fields.

When you speak of appropriation you are also contemplating the cultivation of a new practice in a world. It is like bringing a new vaccine from the lab into the world where widespread inoculation can protect people from the disease.

Yes, I call that the process of *mobilization*. That means to move people in your network into action to get the practice adopted. This is also a large area in which we are doing a lot of thinking and research. Moving people in a network into new action requires more than a skill at reading and sensing concerns. It also requires a sensibility about their histories, concerns, moods, and practices. If you exhibit the sensibility, you will generate trusting relationships, and that will move people. If you see more than one path, choose the one that alleviates the sharpest pain being experienced by the people. Your vaccine will not be adopted if people do not trust you or the results it produces.

Your metaphors of navigating and surfing then refer to a variety of practices for appropriating and mobilizing?

Navigating has to do with a finding a sense of direction for your actions and mobilizations. Many people already understand "navigating the Web" as going from one place to another without a clear idea of exactly how. The Web skill of navigation exemplifies temporality, horizons of concerns, connections between worlds, relevance, traditions, and dynamic discovery of openings for possible actions—the same things that an innovator has to be sensitive to.

Navigating the spaces of possibilities between social worlds is very similar. The possibilities are contingent on other possibilities and they depend on the historical evolution of past possibilities. In navigating we follow connections that move toward the possibilities we want. We are influenced by our estimates of probabilities of success in following a path. Each choice affects our identity and the new possibilities we will see next.

Possibilities are not static spaces, they are dynamic waves. Waves can show up as ideas gone viral, such as rallying slogans in recent international protests about police and terrorism. They can appear as moods of agreement manifested in conversations, showing what people are inclined to follow or resist; for example, companies like Uber and Airbnb are riding waves of interest in the sharing economy and they are discovering that many existing institutions such as municipal regulators and unions are resisting their innovations. Some waves appear as technology and demographic trends that can be extrapolated reliably.

Do innovation stories help people appreciate the new story of innovation?

Yes, indeed. We like to retell some of the familiar stories to see how the new interpretation plays out. One of our old favorites is the story of Louis Pasteur, who saved France from economic ruin the 1880s by developing an anthrax vaccine. Pasteur was a chemist who believed microorganisms produced chemical reactions that showed up as diseases. He took on the anthrax challenge because he suspected a germ was at work and that with his chemistry and microscopes he could observe it and then find a way to neutralize it. He timed his announcement for the maximum impact on a skeptical science community in France by staging a risky experiment in which unvaccinated sheep got anthrax while vaccinated sheep did not. He built a set of new practices based on his germ hypothesis and timed his introductions of innovations so that the social communities would accept and embrace them.

You can see Pasteur building on existing practices (chemistry and microscopes) and using them to propose solutions to unmet concerns (dying sheep and cows). It appears that Pasteur was able to work with the moods and emotions in the social networks of France to create convincing demonstrations of the effectiveness of his methods. He was able to tap into the currents of social power, and ride waves of change to achieve his goals. We noted earlier that Tim Berners-Lee was not simply a technology inventor, he was a great appropriator and mobilizer. The Steve Jobs story can be read the same way.

As you describe them, these new practices of recognizing contingencies, appropriating, surfing, navigating, and mobilizing seem familiar. I have experienced them all at one time or another. I just did not recognize them as essential to the innovations I was pursuing.

That is right. We are all innovators. C

References

1. Flores, F. Report of Consejo Nacional de Innovación para la Competitividad, *Surfing Towards the Future: Chile on the 2025 Horizon* (2013); http://chile-california.org/surfing-towards-the-future-into-counselor-fernando-floress-vision-for-innovation.
2. Flores, F. *The Impact of IT on Business Communications*. ACM 1997 Conference Video Channel; https://www.youtube.com/watch?v=ohZQa8waFO4.

Peter J. Denning (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of ACM *Ubiquity*, and is a past president of ACM. The author's views expressed here are not necessarily those of his employer or the U.S. federal government.

Copyright held by author.

Association for Computing Machinery

ACM Conference Proceedings Now Available via Print-on-Demand!

Did you know that you can now order many popular ACM conference proceedings via print-on-demand?

Institutions, libraries and individuals can choose from more than 100 titles on a continually updated list through Amazon, Barnes & Noble, Baker & Taylor, Ingram and NACSCORP: CHI, KDD, Multimedia, SIGIR, SIGCOMM, SIGCSE, SIGMOD/PODS, and many more.

For available titles and ordering info, visit: librarians.acm.org/pod





DOI:10.1145/2753172

George V. Neville-Neil

Article development led by acmqueue
queue.acm.org

Kode Vicious
Lazarus Code

No one expects the Spanish Acquisition.

Dear KV,

I have been asked to look into the possibility of taking a 15-year-old piece of open source software and updating it to work on a current system used by my company. The code itself does not seem to be too bad, at least no worse than the code I am used to reading, but I suspect it might be easier to write a new version from scratch than to try to understand code I did not write and which no one has actively maintained for several years. What is the point at which I should decide to ignore this old code and write something new?

Lazarus



Dear Lazarus,

Ah yes, GitHub—or was it SourceForge—where old code goes to die, but never quite does. There are probably more abandoned software projects online now than there are projects that are actually used. One of the problems with access to cheap storage is that software engineers, many of whom are not so secretly also data hoarders, need never delete anything. History now will never be forgotten; more's the pity.

Whether or not to resurrect a piece of software depends on a number of variables, including how soon you need the code, how well the code itself worked in the past, and how easy it will be to mentally ingest what you aim to use. I will not address the time question because we know that everyone needs whatever it is yesterday and arguing with management that writing new code will be faster than reusing extant code is a losing battle, one that leads to meetings where you state your name and the substance you have been abusing ever since you started whatever project it was.

From a technical standpoint, you have a few good tools that can tell you how difficult it will be to resurrect a piece of code, and the first is your compiler. Try to build the code with all the warnings and errors set to their strictest levels. If the code builds without problems under these strictures then one of two things is true. First, the code may have been written by someone who really knew what they were doing and who was quite careful in ensuring their code not only adhered to the current language standard, but also that it used a narrow enough subset of the language that changes in the language did not cause any problems. The second possibility is that your compiler is broken. It is actually quite rare that a 15-year-old piece of code will compile completely without warnings under a newer compiler. The number of warnings and errors generated will give you a first indication of how much work you face.

Does the system have a set of tests that were written for it and checked into the repository? If so, you are in luck, and you should go out immediately and play your local lottery. A lot has been written about testing over the last several decades of software development, some of it by KV, and, from time to time, someone actually builds and maintains a good set of unit tests for their software. If these exist, you should read them first, because test code often gives a much deeper insight to what the system is supposed to do than the comments in the actual code.

You say the code is stored in an online repository. If you have gotten past the first two suggestions here then you should read the commit logs for the files that contain the main data struc-

IMAGE BY JULENOCHEK

tures. If you cannot find a simple set of files that contain the data structures, then it is probably time to start again. Good commit logs tell a lot about what a programmer is thinking when developing code. Since data structures are the central place from which all the rest of the code should flow and to which it should always refer back, the logs for these files are of paramount importance. An example of an acceptable message is:

```
Added the ability to store data in offline files. Included both a file name and a file descriptor in the header data structure. The filename is auto-generated by the tmpfile() API if it is not supplied as an argument or in a configuration file by the user.
```

Commit logs that contain short lines such as "fixed bug" are a warning to stay away. It is fine to see that very rarely, but I expect something a lot more descriptive, such as a bug number or link to a bug tracking database as well as a complete description of what the problem was and how the fix addressed it.

Many older pieces of software depend on older versions of operating system and library APIs. That is not a reason to rewrite the code, as many of those changes are mechanical in nature. Deciding how to address the changes in library APIs is a judgment call on the part of the programmer, but those changes should never be grounds to abandon a piece of software unless the library or API itself simply no longer exists.

I would hope this goes without saying, but I have gotten so used to having to say such things anyway that I will say this here. Before you make any changes to the system, make sure to make a copy of the repo and to also import the repo into a fresh repository before you start working on it. Use a tag or similar construct in your source code control system to indicate where the old code stopped and where any of your changes might begin. Otherwise you will be in the unenviable position of not being able to explain where the code came from to the next person who has to maintain it.

KV

Dear KV,

I have been working for a medium-sized software company for a few years and have been recently "promoted" to software architect. This seems to mean that I do all the coding I used to do, but also have to explain to everyone else on my team what to code. Last month I was asked by management to look at the software from a potential company acquisition and to advise them on whether we should or should not acquire the company based on the software that they have already written. This seems like a straightforward case of code review, but is it?

Acquisitive

Dear Acquisitive,

The first thing to do is to become close friends with the engineers at the company, because your company is about to make them rich and then they are all going to quit and you are going to have to maintain their code, without the benefit of a bonus or stock payout. Also, in my experience, no matter what you say, management will do whatever they want, but as an engineer you have to do your best. So let me see if I can come up with a more useful answer.

You are correct that what you are being asked to do is effectively a code review, but it is one with quite a few strings attached and one that is also being done under very different circumstances than a review that might be done within your company. Think of this code review more as a legal deposition. Legal depositions, I am told, are not like those fun courtroom dramas, where two lawyers ask polite questions trying to trick the person under examination into giving away information demonstrating their guilt. It is more like an inquisition where multiple lawyers get to ask any question they like and in any tone they prefer. In this case, you should act like one of those lawyers. Your job is to get as much information out of the engineers at the target of the acquisition as necessary, because, in three months, you will be maintaining their code. Do not let them fob you off with a big .zip file of code. Make sure you get not only code, but access to their tools, repository, and engineer time. Make one or more of them sit in a room with you for as long as necessary to walk through what the system is, how it is supposed to work, and where that happens in the code.

Also make sure to arrange several after-work meals with the engineering group, and get approval to buy as many drinks you think you will need to get them to tell you where the real bodies are buried in the code. That last suggestion is not meant in jest, it is deadly serious. If you want to get information out of engineers, give them plenty of food and drinks. Bring along at least one other engineer from your company, one who you trust and who can take notes and also play good cop. When you ask a hard question, your good cop is supposed to soften whatever you said so the team you are interviewing actually becomes willing to comply with your requests. I have considered hiring someone to just follow me around playing "good cop" to keep me out of trouble, but I think at this point in my career an entourage is not a good idea.

Finally, do not believe in demos. Demos can and will be faked. Make sure you can poke at the system and code before making any serious recommendation to management. Once you have made the recommendation, be prepared to have to work with the code, because even if you say the system contains a function that when executed will spell the end of all humankind, that may not stop management from deciding that the deal is, indeed, a good one, and then stick you with the problem of making it work.

KV

Related articles on queue.acm.org

Beyond Server Consolidation
Werner Vogels
http://queue.acm.org/detail.cfm?id=1348590

Web Services: Promises and Compromises
Ali Arsanjani, Brent Hailpern, Joanne Martin, and Peri Tarr
http://queue.acm.org/detail.cfm?id=639315

Understanding the Problem
George Neville-Neil
http://queue.acm.org/detail.cfm?id=1180184

George V. Neville-Neil (kv@acm.org) is the proprietor of Neville-Neil Consulting and co-chair of the *ACM Queue* editorial board. He works on networking and operating systems code for fun and profit, teaches courses on various programming-related subjects, and encourages your comments, quips, and code snips pertaining to his *Communications* column.

Copyright held by author.

DOI:10.1145/2753237

Meg Leta Jones

Viewpoint
Forgetting Made (Too) Easy

Considering the implications of digital data removal implementations.

TECHNOLOGY POLICY HISTORY was made on May 13, 2014, when the Court of Justice of the European Union (CJEU) found a right to be forgotten in the aging 1995 European Union Data Protection Directive. The decision made waves, because it granted EU citizens the right to request the removal of links from Google's search index that point to identifying content on the Web. Of course, Google and opponents of the right are not happy, but neither are proponents of the right, like myself, who have found potential in a nuanced, well-constructed right to be forgotten. The problem outlined here (there are others) is that the digital right to be forgotten has yet to be developed by common or civil law. It is a generic right with generic exceptions, which is the way all rights are born, but defining and balancing the right to be forgotten has now been delegated to Google—which is in no position to raise a right and certainly has no interest in doing so.



COLLAGE BY ANDRIJ BORYS ASSOCIATES/SHUTTERSTOCK

Background

Although the CJEU interpreted the Directive to grant a right to be forgotten for search engines, it had yet to do so in the law's almost 25-year history. The 1995 Directive required EU member states to put certain protections in place for their citizens within their own national legal systems. The Spanish Data Protection Agency (AEPD) had granted many citizens the right to be forgotten, finding the retention of and access to personal data in these instances to compromise the fundamental right to data protection and digital of persons. In 2010, the AEPD upheld a complaint filed against Google Spain and Google, Inc. by Mario Costeja González involving a 1998 real-estate auction notice related to his social-security debts published by and housed on a newspaper website. The agency dismissed González's claim against the newspaper explaining its continued publication was "legally justified," but upheld the complaint against Google. The company appealed the AEPD's decision to the highest court in Spain, where it was punted to the highest court in the EU, the CJEU.

The CJEU first determined that Google is a "data controller," because the company determines the purpose and means of processing personal data by finding, indexing, temporarily storing, and making available Web content. If Google's activities qualify it as a data controller, it is unlikely that social-media sites and services, as well as third-party data brokers and services, can avoid being labeled data controllers. The CJEU upheld the right as interpreted and applied by the AEPD under Article 12 of the Directive, which guarantees every data subject the right to rectify, erase, or block the processing of data when it does not comply with the Directive, and under Article 14, which guarantees the right to object at any time on compelling legitimate grounds to the processing of data. As a data controller, Google must comply with the Directive and the 28 member states' various rights to be forgotten.

The Good News

First, the good news is that countries around the world are engaged in an important debate about what kind of communication legal structure will reinforce their distinct cultural values. Which legal balance will make their populations freer, fairer, and more prosperous: one where information may be obscured or even deleted, or one where information may linger for a very long time? More good news is the data was not deleted, merely obscured, and the Directive is being updated with a Regulation that seeks to harmonize data protection across the EU member states. The draft Regulation includes a more explicit right to erasure in Article 17. So, there is still some (albeit very little) hope for something to be done about the bad news.

The Bad News

The bad news is worse than the good news. Implementing the decision effectively creates a user request takedown system for personal information given that no intermediary administrative agency or court must determine that an individual's right to be forgotten has been infringed. Individuals enforce their right to be forgotten directly against a data controller. Although this is expeditious for users, it is overly burdensome for data controllers and will lead to an exorbitant amount of obscured and deleted information.[a]

There is no such thing as a frivolous right to be forgotten request—every single takedown notice could legitimately be taken to court.

The right to be forgotten has few established rules or guidance for handling rival interests like expression and historical records. Although some member states have analog/paper versions of the right (generally invoked by those convicted of a crime, who are seeking to prevent others from referencing them in relation to that criminal past), none have actively taken up converting the right to be forgotten to fit a digital world—certainly not to the extent a data controller can know who has a right to retain or delete what data based on what justifications.

The tragic news is that Google (and other data controllers) has been handed a gavel and ordered to decide what the right to be forgotten is without any guidance as to which interests should trump others, when, and why. Google is far removed from the actual information source and has no reason to know why someone posted or maintained access to certain content. The information source is not involved in the obstruction to their content under this system. Not only does Google now carry the huge burden of guessing what the right to be forgotten means as it is flooded with link removal requests, it must restructure its operations to meet these immense compliance costs.

Prior to the ruling, Google directed unhappy users to a help page that tells them to contact the site or service operator to get their problematic content removed. The page then explained that Google might consider removing links only under rare circumstances, such as those that create identity theft threats, but generally requires a court or administrative order to edit search results. When the company receives a takedown order from a governmental body, it simply verifies the legitimacy and complies. This is true for all legal domains except copyright. When Google is notified of allegedly copyright infringing content, it must remove the content immediately to avoid secondary liability.

Now, Google will have to consider each and every user request for removal to determine whether it is a valid right to be forgotten claim with the only guidance from the CJEU that it must take into account amorphous and jurisdiction-specific values like "irrelevant information" and the "public interest."

Forgetting Made Too Easy

There is no such thing as a frivolous right to be forgotten request—every single takedown notice could legitimately be taken to court. Although some kind of jurisprudence will eventually develop within member states as the rare instances when right to be forgotten determinations make their way through various courts, data controllers that do not have the resources or conviction to fight for their right to retain will, in the meantime, simply delete content. Threats of copyright liability have led to a great deal of non-infringing content being removed from the Internet, yet copyright has the luxury of centuries of fair use jurisprudence to offer guidance to those making removal decisions. Not so with the right to be forgotten. We will see data controllers removing content upon request, not wanting to bother with inevitably having to defend in court the decisions they never wanted to make.

The CJEU decision makes forgetting easy. Forgetting should be possible but not easy. C

a Within the first 31 days, Google received a reported 70,000 requests to remove 250,000 search results linking individuals in the E.U. to information the data subject preferred to be disassociated with. As of April 20, 2015, Google has received 241,963 requests to address 877,322 URLs and removed 308,401 URLs (a rate of 41.5%).

Meg Leta Jones (MA1318@Georgetown.edu) is an assistant professor of Communication, Culture, and Technology at Georgetown University.

Copyright held by author.

DOI:10.1145/2667217

Yannis Papakonstantinou

Viewpoint
Created Computed Universe

Computing crosses cosmology and makes the case for agnosticism.

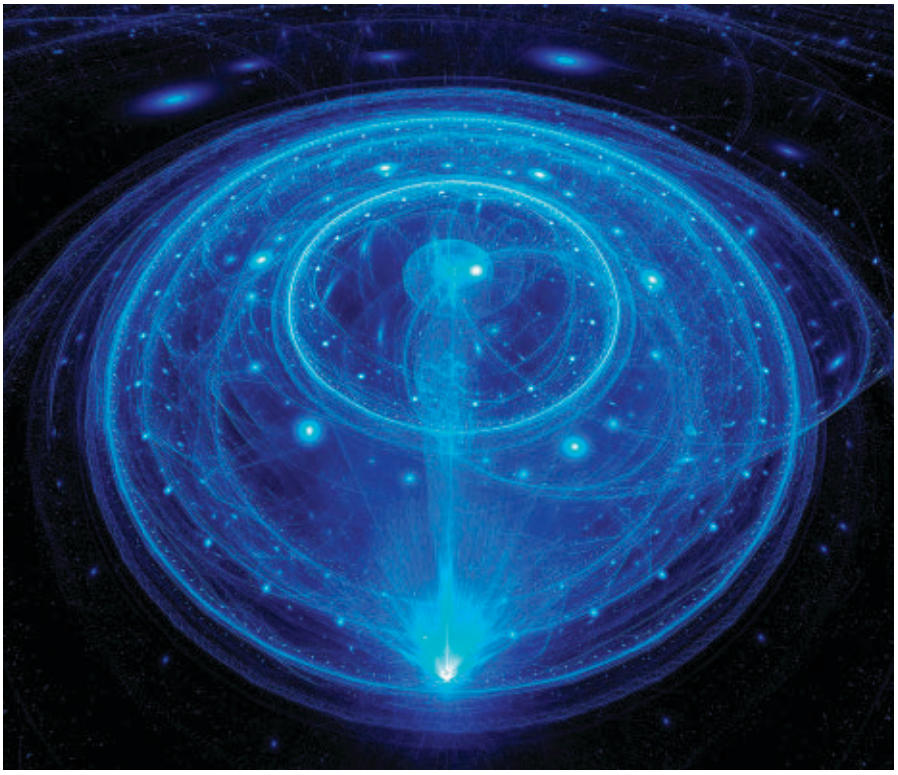


IMAGE BY ATTILA ALEXOVICS

In 1956, just a few years after the birth of computing, Isaac Asimov published a story called the "Last Question."[1] In the story, he envisions a computer-based superintelligence will eventually emerge. The story's highlight is its brilliant ending, in which computing crosses into cosmology, philosophy, and the atheism-creationism debates. Starting from Asimov's story, this Viewpoint discusses how speculation about the future of computing reshapes our perspective on the nature of reality, makes a strong case for agnosticism, and elevates mind and computing into primary cosmological forces.

In Asimov's story, in the year 2150 humanity has solved its energy problems by efficiently harvesting solar energy. A supercomputer, named AC, is the key enabler of this solar energy technology. But two technicians of AC still worry that humanity will eventually die, when the sun and all other energy sources finally exhaust themselves. Since AC has plenty of spare computing cycles it is given the question: How can we keep humanity alive forever? AC replies "INSUFFICIENT DATA FOR MEANINGFUL ANSWER" but keeps thinking on it. Centuries pass and AC becomes a vast superintelligent mind that occupies the mass of whole planets. Still no answer. Myriad years pass, AC becomes a vast superintelligent mind. Every human's mind is uploaded on it. Millions of years pass and AC's computation is a pattern of energy spread out in the whole space, much alike the alien race in Arthur Clarke's *2001: A Space Odyssey* and in the spirit of Ray Kurzweil's prediction[11] that it is the fate of humanity to produce a superintelligent mind, whose computation will eventually consume the matter and energy of vast areas of the universe. Still no answer.

Billions of years pass, AC has thought everything that was worthy of being thought, and there is only this last question remaining. Finally, AC comes up with the answer: *"Let there be light."*

AC becomes the creator of a computed universe. It runs a universe in its computer mind. Had Asimov continued his story, AC's universe would evolve, obtain lands, life, and eventually obtain sentient beings that create their own civilization. The beings would also (re-)invent computing. Indeed, some of these beings might read Viewpoint columns in a magazine plausibly called *Communications of the ACM*.

Is it possible AC has already happened? Could you be one of the sentient beings in AC's created computed universe?

Created Computed Universes

In 1964, the novel *Simulacron 3*,[8] described a universe that is an elaborate computer simulation, yet it includes conscious beings. After abstracting away these stories' details (such as AC's reason for creating a universe) a plausible *Created Computed Universe* (*CCU*) cosmological hypothesis emerges. Mind and computing play a key role in this cosmology, which unfolds in the following three steps: First, we know that mind happens in the cosmos, since we have already seen it happening in our universe. Second, we reasonably speculate that computing technology will enable mind to create CCUs. Then mind can happen in the CCUs as well.[a] Therefore our own universe may be a CCU, created by mind from its parent universe. Note, the physics of a CCU (as perceived by the minds that inhabit it) may be different from the physics of its parent universe. Indeed, any parent universe physics is possible, as long as it enables the emergence of mind.

In 1998, the blockbuster science-fiction movie *The Matrix* brought the CCU idea to the mainstream. In 2003, philosopher Nick Bostrom's simulation argument[2] posited that future civilizations may create CCUs that are simulating the times of their ancestors. Bostrom argued that one must accept one of the following three outcomes:

- Virtually *all* intelligent civilizations (self)destruct before becoming able to launch a simulated universe.
- Virtually *all* intelligent civilizations that can launch a simulated universe, choose not to.
- Almost definitely our universe is simulated, since a primary, non-simulated, intelligent universe can produce a vast number of created universes.

a In a variant, AC designs a physical universe, whose essence is not part of AC's mind. The differences between such a *Created Physical Universe* (*CPU*) hypothesis and the CCU hypothesis do not affect the main conclusions.

Could you be one of the sentient beings in a created computed universe?

CCU, Plato, and Descartes

Is the CCU hypothesis just the computing era's adaptation of a 2,500-year-long series of *skeptical* thought experiments on the nature of reality?[b] Plato's Cave first suggested the true nature of reality may be much different than the perceived reality. 500 years ago Descartes, when he was not busy inventing Cartesian coordinates, was worried about an *Evil Genius*. Descartes writes in the "Meditations on First Philosophy": What do I really know about the nature of reality? My senses could be systematically deceived by an Evil Genius. What I see and what I hear may not be real but just what the Evil Genius feeds to my senses.[c] Some centuries later, the Matrix would become the computerized version of the Evil Genius.

So, the CCU hypothesis appears similar to Plato's Cave and Descartes' Evil Genius: They all suggest there may be a deeper reality below the one we perceive. But there is also a key difference, which is due to the CCU's computing aspect: While until 1950 we would not assign a significant probability to the existence of the Evil Genius (how did the Evil Genius happen?), we now must assign a significant probability to being in a CCU since it appears technologically possible.

b Plato's Cave and Descartes' Evil Genius are two of the many thought-provoking classic ideas and topics the CCU naturally brings forward, along with Occam's Razor, the singularity, digital physics and others. As the author has evidenced at UC San Diego's freshman seminar series "Computing and the Universe," the computing connection of the CCU makes these topics more approachable and exciting to students.

c Descartes grew less worried when he discovered something that is definitely real: *I think therefore I exist*.

CCU and the Atheism-Creationism Debate

Is there a Creator God? This old question recently underlies plenty of polarized debates between biblical creationists and new atheists. The CCU hypothesis bolsters a strongly agnostic position, in the middle between the two camps. Asimov makes it clear that AC is a God creator for its CCU—albeit a material God who is itself the product of evolution.

Notice how the CCU hypothesis deconstructs classic arguments of the atheist camp. Biologist and atheist Richard Dawkins appeals to Occam's Razor when he argues the probability of a creator is extremely small if everything can be explained without assuming a creator.[6] Occam's Razor is the heuristic according to which we should always make the fewest possible assumptions when explaining phenomena.[d] Science lives by it. However, the CCU hypothesis tells us that creators may *emerge*—they are not independent assumptions. Hence, Occam's Razor does not prove very low probability of a creator. Similarly, the infinite regress argument (aka "turtles all the way down"[9]) fails: According to the CCU a creator, with the requisite computing ability, may emerge in a non-created universe.

Darwin's biological evolution boosted atheism by showing that the emergence of life and humans can be explained without positing a creator. Now computing calls us to ponder another kind of evolution, that of computation and mind, and the result is a boost to agnosticism, which can become a positive force. Vocal new atheist thought-leaders have unnecessarily coupled science with atheist assertions (for example, Dawkins[6]). Their dismissals of philosophy's metaphysical questions are also frequent.[3] A boost to rational agnosticism can remove the purported

d Dawkins implicitly adopts a particular interpretation of Occam's Razor: If assuming an entity is not necessary then the probability of its *existence* is very low. Instead of this interpretation, one may adopt a purely methodological one (see survey in the Stanford Encyclopedia of Philosophy[13]). Nonetheless, this Viewpoint adopts Dawkins' interpretation and shows that, even so, Dawkins' use of Occam's Razor does not lead to the very low probability conclusion.

science/metaphysics rift. Practically, once the theory of evolution (and not only) is no more involved into contested metaphysical implications and debates, we can hopefully fully focus on its methodological value and teach it because it produces results.

But are we bound to agnosticism? That is, are we bound to not knowing whether we are or not in a created universe? How about doing some research on it instead?

Are We in a CCU? Digital Physics and Reasons for a Universe

When Konrad Zuse was not busy inventing the first computers,[5] he was speculating that the universe is a giant computer that continuously executes formal rules to compute its next state.[16] Fredkin,[7] Wolfram,[15] and others further contributed on *digital physics*, which posits that the universe is digital in nature and computation makes it tick.

However, the digital physics proponents did not posit a creator of the universe's computation. Nevertheless, the popular press tends to connect digital physics and the CCU. For example, Rundle[12] reports on the experiment that aims to discover if space and time come in tiny, discrete interacting pieces and considers it as evidence the universe is simulated on a digital computer. Although being in a digital universe would lend more credence to the CCU hypothesis, it fails to clearly imply it. Zuse may be right even if we do not live in a CCU. Vice versa, a CCU need not have the telltale signs of the primitive computer simulations of the 21st century.

If digital physics experiments are not much help in settling the CCU hypothesis, how else would we obtain hints? A favorite tactic is to second-guess whether someone who *could* be a universe creator *would* actually do it. Bostrom's argument talks of simulations where the creators make universes in order to simulate the times of their ancestors.[2,e] Vice versa, cosmologist Max Tegmark argues that any superintelligence will withdraw in its own self-contemplation and stay out of universe creation.[14,f]

Are we bound to not knowing whether we are or not in a created universe?

A key problem with creator second-guessing is that it suffers from anthropomorphism, which ironically contradicts the assumption that a creator is a "superintelligent" mind. As *IEEE Spectrum*'s "Singularity" issue[10] summarized, "[after the singularity] the intelligence of ours is no longer the apex of intelligence ... Explaining [the world of a superintelligence] to one of us would be like trying to explain our world to a monkey." Similarly, the reasons for which a superintelligence will or will not create CCUs are most likely incomprehensible by our thinking. For now, agnosticism is the only sure bet: we may be or may not be in a CCU and, if we are, the reasons are unknown.[g]

Awe and Wonder for Our Place in the Universe

Vinton Cerf inspires us by asking "what if it's us?" who will populate and light the galaxy.[4] Ray Kurzweil goes a step further: A superintelligent mind that will emerge in our merger with advanced computing will end up occupying the galaxy's matter and energy with its computation.[11] These are awesome and inspiring visions of humanity practically reaching deification—via computing!

The CCU hypothesis makes a simple reversal in the superintelligence timeline: The universe-creating superintelligence *may* have already emerged. In an era where pop cosmology books regularly announce the latest explanation to everything under the sun, the CCU's intractable mystery is a call to humility. This may at first seem like a disappointment. Yet, the CCU holds its own awe, wonder, and inspiring messages.

The unknown in the CCU is where its magic lies. It can restore the mystical awe and wonder that you felt looking into a dark sky full of stars when you were a child. What is really out there? Unknown and mysterious possibilities. Most inspiringly, a cosmos where computing and mind are at the center.

Imagine what is our place in this cosmos. We may be on the path to deification by advancing our recently discovered computing technology. Or we may already exist in the thoughts and computations of an AC, as Asimov suggested. Or both. No matter what is your choice, remember the words of the great visionary Clarke:[5] *The truth will be far stranger.*

e The students in UCSD's "Computing and the Universe" freshman seminar have their own reasons for a CCU: Imagine a CCU created by an advanced AI civilization, which uploads to the CCU its young and growing-up AI minds to be trained and obtain experience. Or, imagine "reality TV" CCUs—think of the whole universe as a massive *Truman Show*.

f In the same vein, UCSD students (curiously, the vast majority female) believe that a superintelligence will not create a CCU, since evil and pain would probably be part of any non-trivial CCU.

g However, the reasons for a CCU are not necessarily completely unknowable. A future brilliant argument may predict past the singularity horizon. Furthermore, certain assumptions can change the probabilities. For example, an anonymous reviewer points out that if a creator's reasons for a CCU carry over to the created creators, then certain probabilities (for example, universes with no CCUs) decrease.

References

1. Asimov, I. The last question. *Science Fiction Quarterly*, 1956; http://www.thrivenotes.com/the-last-question/.
2. Bostrom, N. Are you living in a computer simulation. *Philosophical Quarterly 53*, 11 (2003), 243–255; http://www.simulation-argument.com/simulation.html.
3. Caroll, S. Physicists should stop saying silly things about philosophy; http://www.preposterousuniverse.com/blog/2014/06/23/physicists-should-stop-saying-silly-things-about-philosophy/.
4. Cerf, V. What if it's us? *Commun. ACM 57*, 3 (Mar. 2014), 7.
5. Clarke, A. *2001: A Space Odyssey.* New American Library, 1968.
6. Dawkins, R. *The God Delusion.* Bantam Books, 2006.
7. Fredkin, E. An introduction to digital philosophy. *International Journal of Theoretical Physics 42*, 2 (Feb. 2003).
8. Galouye, D. *Simulacron-3.* Phoenix Pick, 1964.
9. Hawking, S. *A Brief History of Time.* Bantam Books, 1998.
10. *IEEE Spectrum.* Special Report: The Singularity, 2008; http://spectrum.ieee.org/static/singularity.
11. Kurzweil, R. *The Singularity Is Near: When Humans Transcend Biology.* Penguin Books, 2005.
12. Rundle, M. Physicists may have evidence universe is a computer simulation, 2012; http://www.huffingtonpost.co.uk/2012/10/11/physicists-may-have-evide_n_1957777.html.
13. Stanford Encyclopedia of Philosophy "Simplicity"; http://plato.stanford.edu/entries/simplicity/.
14. Tegmark, M. *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality.* Knopf, 2014.
15. Wolfram, S. A new kind of science. *Wolfram Media*, 2002.
16. Zuse, K. Calculating space. MIT Technical Translation AZT-70-164-GEMIT, http://www.mathrix.org/zenil/ZuseCalculatingSpace-GermanZenil.pdf.

Yannis Papakonstantinou (yannis@cs.ucsd.edu) is a professor of computer science and engineering at the University of California, San Diego.

The author thanks Victor Vianu for his extensive comments and editing. Thanks also to the anonymous reviewers, and to Yannis Katsis, Lamprini Thoma, Menelaos Tzafalias, Vasilis Vassalos, and the students of UCSD's "Computing and the Universe" seminar series for their many comments and suggestions. Finally, thanks to Moshe Vardi for the invitation to write this Viewpoint.

Copyright held by author.



DOI:10.1145/2757637

Vinton G. Cerf

Interview

An Interview with U.S. Chief Technology Officer Megan Smith

Recently appointed U.S. CTO Megan Smith discusses her evolving governmental role.

MEGAN SMITH, AN MIT-trained mechanical engineer and a former vice president in the leadership team of Google X, was appointed U.S. chief technology officer by President Obama in September 2014. With many years of entrepreneurial experience in Silicon Valley, Smith's role as CTO in the Office of Science and Technology has her serving as an assistant to the president, focusing on technology policy, data, and innovation in the U.S. ACM Past President Vinton Cerf recently interviewed Smith in her Washington, D.C. office; the interview presented here has been edited for clarity and length.

I had the most amazing opportunity to meet with Megan Smith at the headquarters of the U.S. Digital Service, just down the block from the White House. Megan was one of the spark plugs of mergers and acquisitions at Google as well as being involved in Google's educational activities among many other things. I wanted very much to hear from her directly about her work in the Office of Science and Technology Policy and am grateful for the time she was able to spare for this interview.

—Vinton Cerf



IMAGE BY JOI ITO

VINTON CERF: I am speaking with Megan Smith, who is now the chief technology officer of the United States. Megan, my first question is what key objectives do you bring to this task at the White House?

MEGAN SMITH: The chief technology officer's job is a position that President Obama started on his first day. It is embedded in the Office of Science and Technology Policy, together with Dr. John Holdren and his team, who is the incredible science advisor that I have the honor of working with. The idea is to advise the President and his team on how to harness technology, innovation, and data on behalf of the American people. So that's a specific thing, and yet a broad thing. And it can evolve, based on what's happening in our nation and what's happening in tech.

In the beginning, the team focused a lot on technology policy, which we continue to do, and on how do we go about bringing a technical team to the group that's doing policy work and listening to the various constituents and the things that are happening. In this area we have all of the topics of the day. Patent reform, copyright, big data, pri-

vacy, student privacy, the Consumer [Privacy] Bill of Rights—these are the kinds of topics we are working on, being part of the broader policy team as the technical group at the table.

That's the first area?

Yes! Our second area is really digital and open government. We're here at the headquarters of the U.S. Digital Service (USDS), and they are such a great example of this effort.

This is the country that not only created the Internet—with the amazing work of you and your colleagues—but it is also the country that created Amazon, Twitter, Facebook, and all of these wonderful companies. We have this talent in the American people, and we need their talent in the American government. It's our government, and whatever we do will be what we do, and it's just that simple.

There's this idea of TQ [Technology Quotient]—like an IQ or EQ. Let's have TQ, technical skills inside government. And let's not just have it in our incredible places like NASA, or NOAA, or the amazing technical team at the Department of Energy. Let's also have it in the White House at the policy table. And let's have it in the digital services team, and let's lift these extraordinarily talented IT and other professionals who are already in government, and get the design thinkers, and the product managers, and the tech folks from industry to come and start collaborating, and make amazing products on behalf of the American people. So that's a big part of this second area, which also includes open government.

The president has been a real leader in getting data out. More than 125,000 datasets have been opened up in his administration. You know we have a long history as a country [of opening data]. Government holds a lot of value it can release so that other players can act. Lots of that opportunity—open government, the open government partnership, national action plans—things the president is leading that we can be a part of.

Our third area is called Innovation Nation. Our country is full of talent. Some people are ending up in high innovation jobs, or they're able to be in schools where they're able to do great STEM [science, technology, engineering, and mathematics] programs, and have really solid education that lifts the youth straight into high-passion jobs that are interesting to them, that matter in the world, and that deliver great products, goods, and fulfill social missions.

But some people are not getting access [to these areas], and we want to fix that. What is interesting is that there is so much talent in most American cities that live in both of those worlds. So part of this project is less about government having to make some type of new program and much more about us integrating many of our play-space initiatives that some amazing local governments already have on behalf of the American people to actually introduce them to each other in all kinds of contexts so they can lift our country themselves, together, with the support of government in interesting ways.

To be specific, there are these things called Tech Meetups, and there are actually 500 Tech Meetups a day in America. Who knew? Now if you go to Silicon Valley and say, "I'd like to join a startup," and you want to join the tech industry, you'll trip over people. Your taxi driver, a person at a coffee shop, people will be like, "Oh do this, talk to this person," and you're in, right? Because that area is so dense with people plugged into that innovation network.

But if you go to many of our cities, you might not know where to find that community. And yet it's there. It's there in every city. You can see it on the Internet, which is so great. So how do we help more Americans participate in this? Or in startup weekends, or different types of maker festivals—things where there are opportunities for them to begin to join this other community that has better paying jobs. So they can take their passion and apply it as a service or social mission.

One of the things I have been working on is the visibility of technical women and minorities.

That's fabulous. In fact, the way I would try to boil this down into one expression is this is putting policy to work. That's really what you're doing.

Yes! And unlocking talent in all of those sectors.

Speaking of talent and education, I am speaking now for the Association for Computing Machinery, and we are very concerned that there are a number of underrepresented cohorts in computer science. Women and other minority groups—you surely have thought about that. How do we get more of these people interested in STEM?

You know from working in this world for so many years that there used to be a lot more women in computer science until the 1980s. The numbers and statistics say 40% were women in the computing world. In fact, in the history of computing, women were the first computers. It was back when humans calculated, and like the idea of a lawyer or an accountant, people were computers.

I have met some of these women, and they're just incredible people. They went on to be a part of the founding of computing. And then suddenly we went off of a cliff. And we're down, certainly in computer science, but not all the STEM fields are in this position. Some of the more pure science areas—biology and others—are more balanced. But for some reason, in computer science we're staying at 10%, 15%, maybe 20% women at any given time. For some reason this is also true in physics, so we need to understand more about why.

Also people of color are not coming to these fields at scale. There are a lot of things going on. One of the most extraordinary ones is what pure bias was then and what unconscious bias is now. People are not aware of the unconscious bias we all have. Our brain works to make decisions with imperfect information, and we fill in things, so a lot of bias comes in.

One of the things I have been working on is the visibility of technical women and minorities. For example, there

are roughly 16 million programmers in the world. That means there are two to three million women programmers. Why don't we ever see them? That's a lot of people. How do we make sure we're not forgetting to balance the panels at conferences, not forgetting to reach out to people?

One of the research points I learned from Brian Welle, who was my colleague at Google and is an amazing scientist in these areas, is a study that he came across showing that when people are applying for a job, if the job lists 10 characteristics, men will apply if they have at least three of the characteristics on average, and women will apply if they have at least seven of the characteristics on average. That just happens to be in our culture. So what do we do about that as managers or leaders when we're hiring? We need to notice there is a set of people with their hands up, and a set of people without their hands up. We need to consider the whole pool and get the right person, help encourage the women to put their hands up, and to take the time to notice this to help them.

The more we can make people like you visible, the easier it will be to convince others that this is a perfectly sensible technical career.

Totally, and also let's look at the people who have been here. Katherine Johnson was a former computer who joined NASA. She was there when Sputnik went up, and she got pulled into the teams that calculated trajectories. She's an African American woman, and she's 97 years old now. She calculated the trajectories for Alan Shepard. John Glenn wouldn't even fly until Katherine double-checked the math! She stayed up like 12 hours recalculating, and if she got the right number, he would say "Okay, Katherine agrees, I'm going." She was one of the trajectory calculators of the Apollo mission, but she's not in the *Apollo 13* movie.

The other day I came across this tweet about Margaret Hamilton, and I looked at her Wikipedia page. It said she had coined the word "software engineering" and created asynchronous software. She was standing with a pile of code as tall as she was that had been for the Lunar Lander, which she

We have extraordinary talent, and we have great history of amazing innovation.

managed and wrote. Margaret had written the design of the software so it would override, and it would only pay attention to landing. But as the Lunar Lander was landing, somebody had apparently left some switches in the wrong places. All this other stuff was coming at the computer, but it was ignoring that because of her code. The Lunar Lander would have been aborted without her software design, which was the first time it had been done that way.

And so we need to find more people like these people, know about them, get them the awards and the representation, put them in the movies because now our children don't see them in the movies, and don't think they could do these things. Hollywood is now doing these really interesting crime shows, and there are all these female forensic scientists who've been cast. So forensic science is off the charts for women going into that field, because they can see it.

We need the help of Hollywood, especially in children's television. I think it's 15 male computer scientists to women that they're depicting without even realizing it—back to unconscious bias. So it just perpetuates itself.

The last thing I need to add to this area is that a lot of the women I know in computer science tell me about their time as a young person in the 1980s and 1990s. A lot of families back then thought, "We'll just get this personal computer," but they just gave it to their boys. I know someone who had to get the key to her brother's room to get the chance to program. We didn't mean to, but that's what we did as a culture. The visuals, the ads, were framed in this handyman way, and we used this cultural bias.

Calendar of Events

June 1–3
20th ACM Symposium on Access Control Models and Technologies
Vienna, Austria,
Sponsored: ACM/SIG,
Contact: Edgar Weippl,
Email: edgar.weippl@tuwien.ac.at

June 3–5
ACM International Conference on Interactive Experiences for TV and Online Video,
Brussels, Belgium,
Sponsored: ACM/SIG,
Contact: David Geerts,
Email: david.geerts@soc.kuleuven.be

June 4–6
SIGMIS-CPR '15: 2015 Computers and People Research Conference,
Newport Beach, CA,
Sponsored: ACM/SIG,
Contact: Diana L. Burley
Email: dburley@gwu.edu

June 10–12
SIGSIM-PADS '15: SIGSIM Principles of Advanced Discrete Simulation,
London, U.K.,
Sponsored: ACM/SIG,
Contact: Simon JE Taylor,
Email: Simon.Taylor@brunel.ac.uk

June 14–17
STOC '15: Symposium on Theory of Computing (part of FCRC),
Portland, OR,
Sponsored: ACM/SIG,
Contact: Rocco Anthony Servedio,
Email: rocco@cs.columbia.edu

June 15–16
IWQoS '15: International Symposium on Quality of Service 2015,
Portland, OR
Co-Sponsored: ACM/SIG,
Contact: Jun Li
Email: lijun@cs.uoregon.edu

June 15–16
TRANSACT '15: 10th Workshop on Transactional Computing,
Portland, OR,
Sponsored: ACM/SIG,
Contact: Justin E. Gottschlich
Email: justin@nodeka.com

INTERACTIONS

ACM's *Interactions* magazine explores critical relationships between people and technology, showcasing emerging innovations and industry leaders from around the world across important applications of design thinking and the broadening field of interaction design.

Our readers represent a growing community of practice that is of increasing and vital global importance.



To learn more about us, visit our award-winning website
http://interactions.acm.org

Follow us on Facebook and Twitter

To subscribe:
http://www.acm.org/subscribe

Association for Computing Machinery

Many people are coming to government from the tech world, building products and services and ideas.

Bletchley Park is where the incredible team cracked the Nazi enigma codes during World War II and saved 11 million lives. *The Imitation Game* movie is about this team. Yet the visitors to the Bletchley Park museum are 25:1 boy visitors to girl visitors. So even today the British public thinks this museum is for boys and yet, of the 10,000 people who worked at Bletchley Park, 8,000 were women including elite code breakers like Joan Clark. The Duchess of Cambridge's grandmother, it turns out, was a code breaker. So one of things I am hopeful about is that Kate will lead the charge on getting girls to visit the museum along with the boys, because we need everyone to know about these heroes.

Let me shift our conversation a little bit, and let's talk about the United States and competitiveness on a global scale. What do you see we need to do to make ourselves continue to be competitive?

I really believe it's about talent. We have extraordinary talent, and we have great history of amazing innovation, and so we can keep that if we can keep pulling everyone. One of the greatest times was actually during World War II, when a lot of diverse groups sat at the table together to rebuild the Pacific fleet. So sometimes it's about enabling people. When the women and men who rebuilt the Pacific fleet were working in the Bay area, they opened something like 30 daycare centers in three months. We need to be thinking about how we can support parents in our workforce. We're really paying attention to those kinds of important policies, and so I am happy about what the president is doing with early childhood education on that front to enable parents. I am also really happy about the early childhood education work because my mom was a kindergarten and pre-K teacher who's retired now, but I just know from growing up how important those years are. I think three-year-olds' brains are twice as wired as grown-up humans, so the more we can do to make sure those [underprivileged] kids are staying up with other kids the better.

I brought a Raspberry Pi and an Arduino. This is the older Raspberry Pi [holding the unit up]—they just launched the new one. This is, of course, from the U.K.—the old BBC micro team kind of reincarnated into this new one; well, it's really the board from your cellphone. The U.K. is also making sure that all of their children are learning how to play with it, program it, and they're bringing it into school—especially elementary school.

The Arduino is the American one, there's also Intel, Galileo, and bunch of great, great, products here in the U.S. Not only is it cool and a real computer—it can run YouTube and Minecraft and programs.

My friend Dr. Sue Black has second, third, and fourth graders taking the board and plugging in the speaker and the mic. They're learning to ask, "What is this thing, this computer thing?" This is just like Steve Jobs, Wozniak, and Gates got to do when they were young so they became intuitive with these technologies. So I think [it's important to be] doing this so children can have active experiences playing with these things when they're younger. Dr. Black also notes that for second, third, and fourth grades, the parents are not afraid of their homework, as opposed to fifth, sixth, and seventh grade and on. So it's actually a good way to help your 20-, 30-, and 40-year-olds learn about this too, because the second graders can teach their parents.

This must fit well into the Maker movement in some way.

Yes, very much. Arduino and all of these others are doing a kind of Internet of Things. You may think that sometimes these more playful things are not really related to advanced man-

ufacturing's future, but they very much are. As we think about Smart Cities and using advanced computers and sensors to compute, these great tools are here. So I think a combination of focusing on our youth to make sure they feel really fluent in coding and with these kinds of technologies, and also focusing on the adults who are here, who are going to live for many, many years. We have to get scrubbed in on this kind of work, whether it's the actual tech, or being a designer, in marketing or sales, or using this for a social mission—we need all of the talents at the table.

So this field is very much like the kind of virtuous circle, where the introduction of these ideas and technologies going on makes you comfortable with them, and lets you invent with them without fear?

Right, exactly.

And there's no bias, associated with that at all?

I was lucky because we had mandatory science fair when I was a kid, and I went to an inner-city public school in Buffalo, a great school. Because we had to do science fair, everybody did it, and we had a very diverse group of kids who went off to the city science fair, county science fair, and so on. It will work if you get everybody in, and get them early.

Let's see if we can finish up with a couple of other questions Megan. One of them has to do with your experiences in Silicon Valley. You came from a very intense, highly technically oriented place to Washington, D.C., which is not necessarily always known for that. Did you bring any experiences to your job at the White House that you have been able to apply, or have struggled with?

Yes, many, many experiences. We brought a lot; many people are coming to government from the tech world, building products and services and ideas. The teams here are very entrepreneurial. They're similar to us culturally—they just haven't been doing tech entrepreneurship or technology as much. They had been doing more distant contracting for all of that, rather than having [technical expertise] in the room with them.

And still you might contract, and you'll be more informed about how you're doing things, or how you're buying things, or architecting the system with a teammate in the room who knows the stuff.

But really, to do great legislation, to do economics, you have to not only leverage great academy and corporate ideas, but also a level of entrepreneurship to get things moving as a cohort. Sometimes I feel like in the past, government built this amazing car and accidentally left one tire flat. [Cerf laughs out loud.] So it's not that you need all of these [technical] people, you just sometimes need a few to get things going.

And we really have a great team, and I love my colleagues here. We actually have the most diverse group I have ever worked with as a leadership team. There are people from all different parts of America, all age groups, men and women, in a really interested, talented team. I'm happy the president stepped up through some tough experiences like Healthcare.gov. He saw how people fixed it, and he saw that a set of Americans were missing from his team, and he went to collect them. And we're here, and we're happy to be here. It's an honor to be here.

Sometimes I feel like in the past, government built this amazing car and accidentally left one tire flat.

That actually sounds very entrepreneurial. You take some risks, and bring people in to fix things. Let's finish up with a really important question: What advice would you give to young people about what they should be thinking about or doing for the future?

I was really lucky to take acoustics from Professor (Amar) Bose. He was amazing, and he taught us a lot of incredible physics of acoustics, and a little bit about business, and about the things he had been part of. The main thing he emphasized was that you have to find your passion. Find what you love ... and that doesn't mean you'll know right away. There are so many things you have to do in the world, but if you practice a lot of disciplines and try a lot of things, especially as a young person, don't let your bias make you think you can't do this or that or whatever.

You're going to live to 100-something years probably if you're born or young today. So you'll have a lot of chances to do a lot of things, so don't pigeonhole yourself. Try a lot of things, and then as you try them, let your passion steer you.

You have to find what you want to bring to this world. The more you can be doing the things you love, developing your wonder and curiosity, and being in service to other people, the more you can be helping other people, the more you get back in life. So it's sort of combining those things. If I hadn't had incredible teachers who made us do this science and tech stuff—in addition to the other topics of history and writing, which are also incredible and I love—I wouldn't have known about this secret world of innovation. And to be of service using innovation and invention and technology, which can really solve some of our greatest challenges with other colleagues, is an honor.

That's terrific advice. I hope that the kids who read this take your advice, so they can take us all into the future. Thank you so much Megan, we really appreciate your time.

Thank you! And thanks for your incredible leadership, and what you've brought to our world with your innovation, your invention, and also just the way you approach the world. For people who haven't gotten to interact with Vint, he is an amazing colleague who includes everyone, encourages everyone, and can see how big the talent base of this world truly is.

Well I'm just glad to be of service. Thanks, Megan! C

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

Copyright held by author.

DOI:10.1145/2745390

Article development led by acmqueue
queue.acm.org

Lessons learned managing a data science research team.

BY KATE MATSUDAIRA

The Science of Managing Data Science

"WHAT ARE THEY doing all day?"

When I first took over as VP of Engineering at a startup doing data mining and machine learning research, this was what the other executives wanted to know.

They knew the team was super smart, and they seemed like they were working really hard, but the executives had lots of questions about the work itself. How did they know that the work they were doing was the "right" work? Were there other projects they could be doing instead? And how could we get this research into the hands of our customers faster?

And that was my job. You do not hire a VP of Engineering unless you need a VP of Engineering.

Over the next year we addressed all these questions, and we were able to shorten our timeframe from idea to product from six months, to 2–3 months. In the rest of this article I want to share with you some of the lessons we learned managing a data science research team that worked closely with a product team. And I hope they give you some ideas and insights you can integrate into your own work and process.

When you have a team of people working on hard data science problems, the things that work in traditional software do not always apply. When you are doing research and experiments, the work can be ambiguous, unpredictable, and the results can be difficult to measure.

As a manager coming from the standard software development process, I was well versed in agile, scrum, and the other processes du jour. However, the research work we were doing did not fit neatly into those boxes. Things were a bit too messy.

We were proposing ideas, investigating hunches, and testing hypotheses. It was very difficult to estimate the work, and we could not guarantee the results would be good enough to ship. So we had to come up with a new process that would account for all of this uncertainty.

Back to our original question: "What are they doing all day?"

I could have easily gone and just asked everyone and provided a status report, but that would not have really addressed the actual question.

The heart of the matter was that we needed a clear process that would work well for research, while still making collaboration with our product teams easy. We needed a process where we could take advantage of the synergies of a group of people who know a lot about the data and a group of people that know a lot about our customers, and get them working together as one team.

We needed clear channels of communication and a process that would support our future work. And with that in mind, we focused on the key areas of communication and logistics.

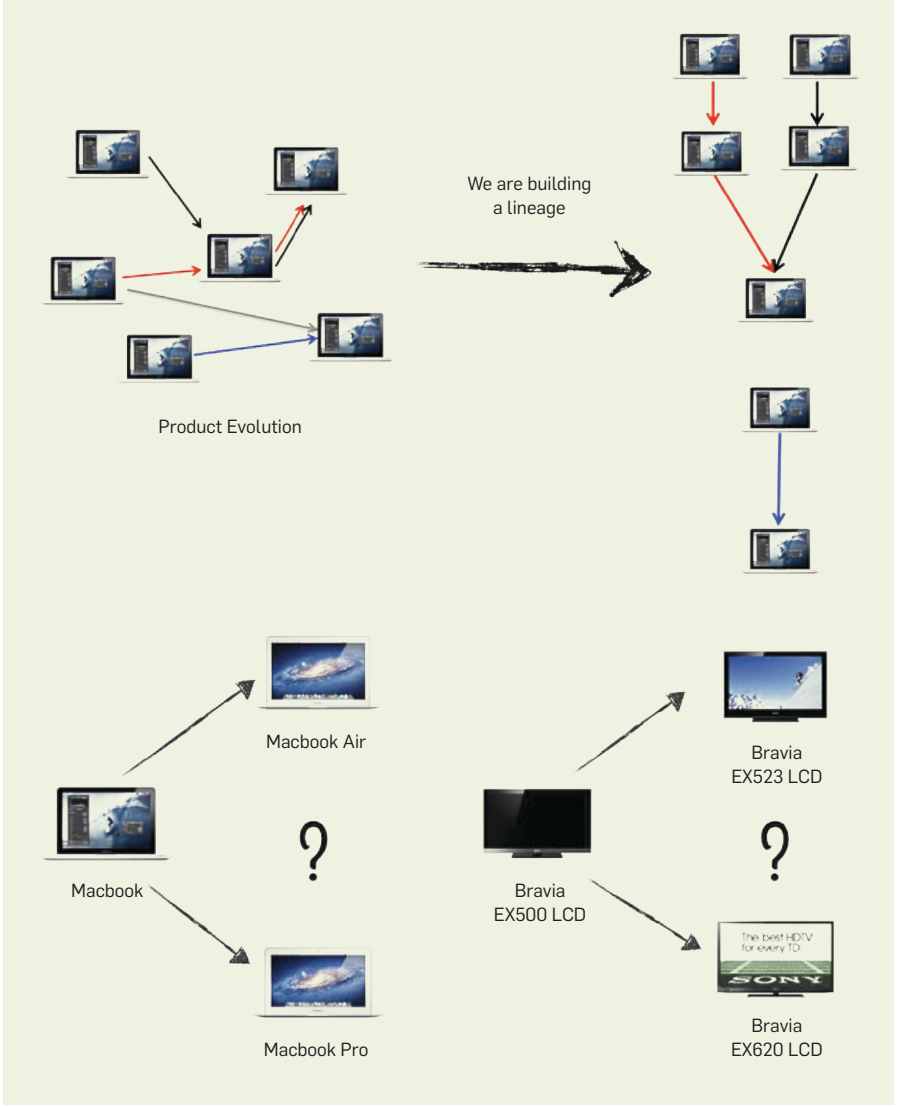
Communication: Bringing Transparency

The very first thing we had to do was improve our communication. Our dif-

ILLUSTRATION BY JUSTINE BECKETT



Model lineage examples.



ferent teams were not speaking the same language.

Answering someone's question was not as simple as just explaining the right answer; it was about learning to use the right vocabulary to reach true understanding. *Communication isn't just about the words, but the meaning behind them.*

How are you communicating results? For example, one thing that always tripped us up was the question "What does it mean to be finished?" The data science team was measuring their results, and we would complete an experiment, but how did we know when an algorithm or model was good enough to be integrated into the product? We needed more than the measurement of results; we needed success criteria.

When we were creating a model, we would measure the precision and recall of an algorithm and communicate our results that way. However, not all executives speak in those terms. So we started changing the way we talked about experiment results. Instead of putting things in terms of algorithm evaluation metrics (like precision and recall), we started communicating results in terms of the customer experience and business metrics.

Before, we would communicate results like:

Precision: 80%
Recall 25%

And after we would communicate results this way:

For the top search terms* accessories appear 25% of the time on the first page.

For top search terms the head products are present 90% of the time in the top 3 results, 98% of the time in the top 5.

* Top search terms are the 1,000 most popular queries on our website over the last 30 days.

The second way made it clear what these measurements meant. And even people without understanding of the algorithms could understand and ask questions about the results.

How are you measuring the results? Another issue we ran into was we were using a random sampling to measure our results for an algorithm or model. And while that is certainly one way of testing the results, it did not necessarily map to the experience on the website. The algorithms we picked would perform well, but then completely miss the boat on an important product.

So we changed the way we evaluated our results by weighting them with actual customer usage. We would take head products, or products that had customer views, and then use those as the samples to measure our results. This gave us a better approximation of the real customer experience.

Show improvements. Another key when you are doing experiments to improve an algorithm, like tweaking search results, is to show the before and after. What changed with the experiment? Sometimes certain results would be improved, and others would decline, so it was equally important to understand and communicate both sides of the coin. With all the information we could make the best decision about how to proceed.

Solving difficult problems. Besides all the problems with communicating about the data and results, another miscommunication seemed to happen when we would embark on new ideas. The conversation would go something like:

An executive would say "I have this idea to create a model lineage for a product. Then we can show customers past products and use our algorithms to predict when new ones will be released."

A data scientist would take the requirements and go investigate the idea, work on it for a while and come

back with: "That is going to be really hard; I don't think we should do it."

But these conversations were not always consistent.

A lot of things were "really hard" and sometimes features would get pushed through anyway (frustrating the data science team that had to work on them), but other times something that was "really hard" would just stop the feature in their tracks—resulting in missed opportunity.

Since there was not a way to follow up on these conversations, or problem-solve as a group, hard problems only became harder.

If you are working in data mining and machine learning research, most of the problems are difficult. The thing is, sometimes the reason why they are hard can be solved in other ways—like better/different data, changing requirements, or adding special cases. So saying a problem would be "really hard" wasn't a good enough reason not to try to solve it; and yet, it was consistently being used as the reason why things were not happening.

The key was learning to communicate why things were difficult.

For us, the thing that worked best was to use analogies and examples. By getting specific about why it was difficult, it was easy for people to understand the challenges. So for model lineages we used the examples in the accompanying figure.

All of these changes were just about thinking how we were communicating. It was tailoring the message for the audience. If you work in data science or research, it can be easy to forget that not everyone is as immersed in the details as you are, and that you need to communicate things a bit differently to ensure understanding.

Logistics: Establish a Clear Process

The next step was adjusting how we did the work. Once we had addressed some of the communication problems, I can remember someone asking me, "How can we create a sense of urgency for the research team?" This comment wasn't made because work wasn't getting done, but simply because the research team did not really have deadlines. The work was finished when it was ready.

But we were a startup; everyone else in the company had deadlines, and we were racing against the clock to deliver as much value as possible while we had the runway to do it.

Adding deadlines to research. Research is called research for a reason, and even if we wanted to have a schedule and wrap up the deliverables with a bow, that does not mean the world would work that way. What we did, though, was create a backlog for our experiments, just like a backlog of tasks in traditional software development. These were all the things we wanted to try, in priority order.

For each of these items, we were able to define success criteria, assign them out, and even cost them (how long should this take?).

This ended up being really beneficial for the team. Instead of each person working on their own on one thing, we became a team working together from the same list of priorities.

Adding agile demos. Another thing we did was we scheduled monthly demos, where each member of the research team would share their progress. These heartbeat meetings helped keep everyone on the same page, and helped show the real progress that was being made.

These touch points also helped create a sense of urgency (since the demos created a deadline) and all of sudden we were moving much faster as a team.

Since we were still dealing with the unpredictable nature of research (who knows if an experiment would work or how long it might take to run and tweak the results), we decided to place a focus on speed.

We wanted quick proof-of-concept prototypes and fast iteration. Previously we might have waited until we had concrete conclusions to show results to the product teams, but in this new model we would share data and intermediate results more readily. We would just add lots of caveats.

At first this made the team uncomfortable; they did not want to show things they knew they could still improve. However, once we started doing this more regularly we were able to get feedback even earlier in the process. This early feedback even helped us make adjustments to the product along the way (since we were able to see how the results actually looked, which was sometimes a lot different than we had expected or mocked up in the designs).

And when the product team started seeing the results, they would get really excited, which fueled the research team to get even more done.

For example, one of the features we wanted to work on was value estimation, where we would take apart the features of a product and assign a value to each one. The goal was to help users compare different products based on the features so they could understand what they were really paying for, and discover cheaper products with similar specifications.

Well after doing a lot of work, it was clear this was a really hard problem. However, using the same data and models, we were able to build a feature that allowed them to accomplish the same goals—comparing products, instead of features. C

Related articles on queue.acm.org

How Will Astronomy Archives Survive the Data Tsunami?
G. Bruce Berriman and Steven L. Groom
http://queue.acm.org/detail.cfm?id=2047483

Privacy, Anonymity, and Big Data in the Social Sciences
Jon P. Daries et al.
http://queue.acm.org/detail.cfm?id=2661641

Queue Portrait: Hilary Mason
http://queue.acm.org/detail_video.cfm?id=2532200

Kate Matsudaira (katemats.com) has just recently made a foray into novel territory as founder of her own company Popforms. Prior to that she worked in engineering leadership roles at companies such as Decide (acquired by eBay), Moz, and Amazon.

Copyright held by author.
Publication rights licensed to ACM. $15.00

DOI:10.1145/2732629

Article development led by acmqueue
queue.acm.org

...or: How I stopped worrying and learned to love time.

BY ŠTĚPÁN DAVIDOVIČ AND KAVITA GULIANI

Reliable Cron Across the Planet



THIS ARTICLE DESCRIBES Google's implementation of a distributed Cron service, serving the vast majority of internal teams that need periodic scheduling of compute jobs. During its existence, we have learned many lessons on how to design and implement what might seem like a basic service. Here, we discuss the problems that distributed Crons face and outline some potential solutions.

Cron is a common Unix utility designed to periodically launch arbitrary jobs at user-defined times or intervals. We will first analyze the basic principles of Cron and its most common implementations and then review how an application such as Cron can work in a large, distributed environment, in order to increase the reliability of the system against single-machine failures. We describe a distributed Cron system that is deployed on a small number of machines, but is capable of launching Cron jobs on machines across an entire datacenter, in conjunction with a datacenter scheduling system.

Before we jump into the topic of how to run a reliable Cron service for entire datacenters, let us first check out Cron—introduce its properties and look at it from the perspective of a site reliability engineer.

Cron is designed so that both system administrators and ordinary users of a system can specify commands to be run and when to run them. Various kinds of jobs, including garbage collection and periodic data analysis, are executed. The most common time specification format is called "crontab." It supports simple intervals (for example, "once a day at noon" or "every hour on the hour"). Complex intervals such as "every Saturday, and on the 30th day of the month" can also be configured.

Cron is usually implemented using a single component, commonly called `crond`. This is a daemon that loads the list of Cron jobs to be run and keeps them sorted based on their next execution time. The daemon then waits until the first item is scheduled to be executed. At that time, `crond` launches the given job or jobs, calcu-



lates the next time to launch them, and waits until the next scheduled execution time.

Reliability

Viewing the service from a reliability perspective, there are several things to notice. First, the failure domain is essentially just one machine. If the machine is not running, neither the Cron scheduler nor the jobs it launches can run. (Failure of the individual jobs, for example because they cannot reach the network or they try to access a broken HDD, is beyond the scope of this analysis.) Consider a very simple distributed case with two machines, in which the Cron scheduler launches jobs on a separate machine (for example, using SSH). We already have distinct failure domains that could impact our ability to launch jobs: either the scheduler or the destination machine could fail.

Another important aspect of Cron is the only state that needs to persist across `crond` restarts (including machine reboots) is the crontab configuration itself. The Cron launches are fire-and-forget, and `crond` makes no attempt to track them.

A notable exception to this is `anacron`, which attempts to launch jobs that were scheduled to be launched when the system was down. This is limited to jobs that run daily or less frequently, but is very useful for running maintenance jobs on workstations and notebooks. Keeping a file containing timestamps of the last launch for all registered Cron jobs facilitates running these jobs.

Cron jobs and idempotency. Cron jobs are used to perform periodic work, but beyond that, it is difficult to know their functions in advance. Let us digress for a bit, and discuss the actions of Cron jobs themselves, because understanding the variety of requirements of Cron jobs will be a theme throughout the rest of the text, and obviously affects our reliability requirements.

Some Cron jobs are idempotent, and in case of system malfunction, it is safe to launch them multiple times—for example, garbage collection. Other Cron jobs should not be launched twice, for example, a process that sends out an email newsletter with wide distribution.

To make matters more complicated, failure to launch is acceptable for some Cron jobs but not for others. A garbage collection Cron job scheduled to run every five minutes may be able to skip one launch, but a payroll job scheduled to run once a month probably should not.

This large variety of Cron jobs makes it difficult to reason about failure modes: for a service like Cron, there is no single answer that fits every situation. The approach we will be leaning toward in this article is to prefer skipping launches, instead of risking double launches, as much as the infrastructure allows. The Cron job owners can (and should!) monitor their Cron jobs, for example, by having the Cron service expose state for the jobs it manages, or by setting up independent monitoring of the effect of the Cron jobs. In case of a skipped launch, Cron job owners can take action that matches the nature of the job. However, undoing a double launch may be difficult, and in some cases (consider the newsletter example) impossible. So,

IMAGE FROM SHUTTERSTOCK.COM

we prefer to "fail closed" (in engineering terminology) to avoid systemically creating bad state.

Cron at Large Scale

Moving away from single machines toward large-scale deployments requires some fundamental rethinking of how to make Cron work well in such an environment. Let us discuss those differences and the design changes they necessitated, before presenting the details of Google's Cron solution.

Extended infrastructure. Regular Cron is limited to a single machine. However, for large-scale system deployment, our Cron solution must not be tied to a single machine.

If we assume a single datacenter with exactly 1,000 machines, a failure of just 1/1,000th of the available machines would knock out the entire Cron service. For obvious reasons, this is not acceptable.

To solve this general problem, we decouple processes from machines. If you want to run a service, simply specify which datacenter it should run in and what it requires—the datacenter scheduling system (which itself should be reliable) takes care of figuring out which machine or machines to deploy it on, as well as handling machine deaths. Launching a job in a datacenter then effectively turns into sending one or more RPCs to the datacenter scheduler.

However, this process is not instantaneous. There are inherent delays associated with both discovering a dead machine (health-check timeouts) and rescheduling a job onto a different machine (software installation, startup time of the process).

Since moving a process to a different machine can mean loss of any local state stored on the old machine (unless live migration is employed), and the rescheduling time may exceed the smallest scheduling interval of one minute, we should be able to cope with this situation. One of the most obvious options is to simply persist the state on a distributed file system such as GFS, and use it during startup to identify jobs that failed to launch due to rescheduling. This solution falls short of timeliness expectations, however: if you run a Cron job every five minutes, a delay of a minute or two caused by the total overhead of rescheduling is a substantial portion of the interval.

The timeliness expectation might motivate keeping around hot spares, which could quickly jump in and resume operation.

Extended requirements. Another substantial difference between deployment in a datacenter and on a single machine is how much more planning is required when it comes to resources such as CPU or RAM.

Single-machine systems typically just co-locate all running processes with limited isolation. While containers (such as Docker[3]) are now quite common, it is not necessary or common to use them to isolate absolutely everything, including `crond` and all the jobs it runs, for software deployed on a single machine.

Deployment at datacenter scale commonly means deployment into containers that enforce isolation. Isolation is necessary because the base expectation is that a running process in the datacenter should not negatively impact any other process. In order to enforce that, you need to know what resources to acquire upfront for any given process you want to run—including both the Cron system and the jobs it launches. As a logical consequence of this, a Cron job may be delayed if the datacenter does not have available resources matching the demands of that job. This, as well as the desire to monitor the Cron job launches, means we must track the full state of our Cron jobs, from scheduled launch through termination.

Because the Cron system has now decoupled the process launches from specific machines, as described earlier, we may experience partial launch failures. Thanks to the versatility of the job configuration, launching a new Cron job in a datacenter can necessitate multiple RPCs. This, unfortunately, means we can reach the point where some RPCs succeeded, but others did not, for example, because the process sending them died in the middle. The recovery procedure has to handle these as well.

In terms of failure modes, a datacenter is substantially more complex than a single machine. The Cron service that started as a relatively simple binary on a single machine now has many obvious and less obvious dependencies in the datacenter case. For a service as basic as Cron, we want to ensure the service still functions even if the datacenter suffers a partial failure (for example, a partial power outage or problems with storage services). To improve reliability, we ensure the datacenter scheduler locates replicas in diverse locations within the datacenter, to avoid, for example, a single power distribution unit taking out all the Cron processes.

It might be possible to deploy a single Cron service across the globe, but deploying Cron service within a datacenter has benefits such as shared fate with the datacenter scheduler (which is the core dependency) as well as low latency to it.

Cron at Google and How to Build One

Let us address the problems we must resolve in order to provide a reliable Cron in a large-scale distributed deployment and highlight some important decisions made for distributed Cron at Google.

Tracking the state of Cron jobs. As discussed, we should hold some amount of state about the Cron jobs and be able to restore it quickly in case of failure. Moreover, the consistency of that state is paramount: it is more acceptable to not launch Cron jobs for a while than to mistakenly launch the same Cron job 10 times instead of once. Recall that many Cron jobs are not idempotent, for example, a payroll run or sending out an email newsletter.

We have two options: store data externally in a generally available distributed storage, or store a small amount of state as part of the Cron service itself. When designing the distributed Cron, we opted for the second option. There are several reasons for this:

- Distributed file systems such as GFS and HDFS are often designed for the use case of very large files (for example, the output of web crawling programs), whereas the information we need to store about Cron jobs is very small. Small writes on such a distributed file system are very expensive and have high latency as the file system is not optimized for them.
- Base services with wide impact, such as Cron, should have very few dependencies. Even if parts of the datacenter go away, the Cron service should be able to function for at least some amount of time. This does not mean the storage has to be part of the Cron process directly (that is essentially an implementation detail). However, it should be able to operate independently of downstream

systems that cater to a large number of internal users.

The Use of Paxos

We deploy multiple replicas of the Cron service and use the Paxos algorithm to ensure consistent state among them.

The Paxos algorithm[5] and its successors (such as Zab[4] or Raft[7]) are commonplace in distributed systems nowadays (Chubby[1] and Spanner,[2] among others). Describing Paxos in detail is beyond the scope of this article, but the basic idea is to achieve a consensus on state changes among multiple unreliable replicas. As long as the majority of Paxos group members are available, the distributed system, as a whole, can successfully process new state changes despite the failure of bounded subsets of the infrastructure.

The distributed Cron uses a single master job, shown in Figure 1, which is the only replica that can modify the shared state, as well as the only replica that can launch Cron jobs. We take advantage of the fact the variant of Paxos used, known as Fast Paxos,[6] uses a master replica internally as an optimization—the Fast Paxos master replica also acts as the Cron service master.

If the master replica dies, the health checking mechanism of the Paxos group discovers this quickly (within seconds); with other Cron processes already started up and available, we can elect a new master. As soon as the new master is elected, we go through a master election protocol specific to the Cron service, which is responsible for taking over all the work left unfinished by the previous master. The master specific to Cron is the same as the Paxos master, but the Cron master must take additional action upon promotion. The fast reaction time for electing a new master allows us to stay well within a generally tolerable one-minute failover time.

The most important state we keep with Paxos is information about what Cron jobs get launched. For each Cron job, we synchronously inform a quorum of replicas of the beginning and end of each scheduled launch.

The roles of the master and the slave. As noted, our use of Paxos and its deployment in the Cron service has two assigned roles: the master and the slave. Let us go through the operation of each role.

The master replica is the only replica actively launching Cron jobs. The master has an internal scheduler which, much like the simple `crond` described at the beginning, maintains the list of Cron jobs ordered by their scheduled launch times. The master replica waits until the scheduled launch time of the first element of this list.

When we reach the scheduled launch time, the master replica announces it is about to launch this particular Cron job, and calculates the new scheduled launch time, just as a regular `crond` implementation would. Of course, as with regular `crond`, a Cron job launch specification may have changed since the last execution, and this launch specification must be kept in sync with the slaves as well. Simply identifying the Cron job is not enough: we should also uniquely identify the particular launch using the start time, to avoid ambiguity in Cron job launch tracking (especially likely with high-frequency Cron jobs, such as those running every minute). This communication is done over Paxos. Figure 2 illustrates the progress of a Cron job launch from the master's perspective.

It is important to keep this Paxos communication synchronous and not proceed with the actual Cron job launch until there is a confirmation that the Paxos quorum has received the launch notification. The Cron service needs to know whether each Cron job has launched, to decide the next course of action in case of master failover. Not performing this synchronization could mean the entire Cron job launch happened on the master, but the slave replicas were not aware of it. In case of failover, they might attempt to perform the same launch again simply because they were not informed it had already happened.

The completion of a Cron job launch is announced via Paxos to the other replicas, also synchronously. Note that it does not matter whether the launch succeeded or failed for external reasons (for example, if the datacenter scheduler was unavailable). Here, we are keeping track of the fact the Cron service has attempted the launch at the scheduled time. We also need to be able to resolve failures of the Cron system in the middle of this operation, which we discuss later.

Another important feature of the master is as soon as it loses its mastership for any reason, it must immediately stop interacting with the datacenter scheduler. Holding the mastership should guarantee mutual exclusion of

Figure 1. The interactions between distributed Cron replicas.

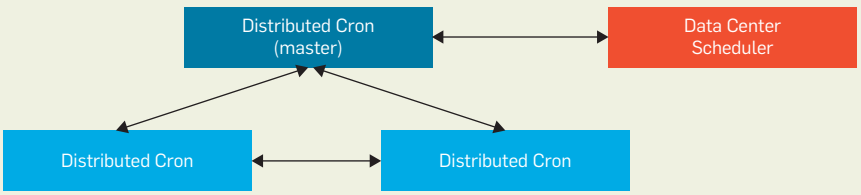


Figure 2. Progress of a Cron job launch.

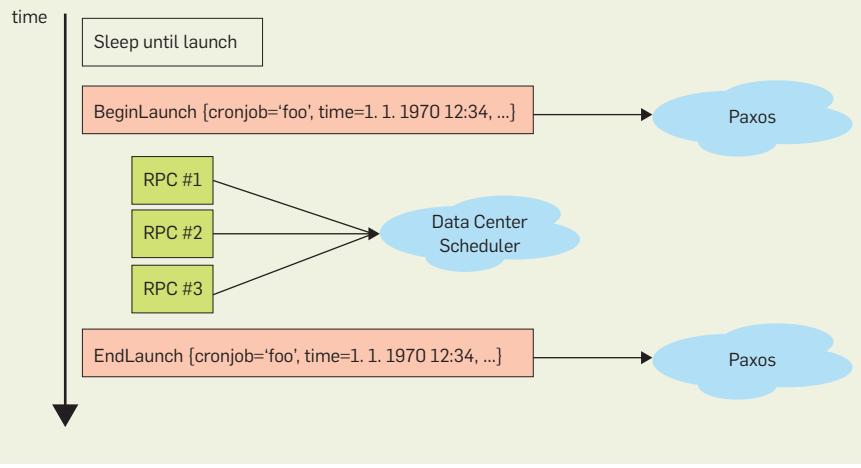
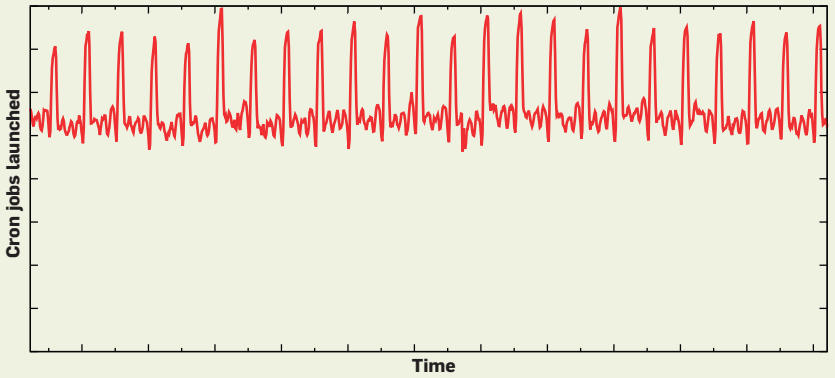


Figure 3. The number of Cron jobs launched globally.



access to the datacenter scheduler. If it does not, the old and new masters might perform conflicting actions on the datacenter scheduler.

The slave replicas keep track of the state of the world, as provided by the master, in order to take over at a moment's notice if needed. All the state changes that slave replicas track are coming, via Paxos, from the master replica. Much like the master, they also maintain a list of all Cron jobs in the system. This list must be kept consistent among the replicas (through the use of Paxos).

Upon receiving notification about a launch, the slave replica updates its local next scheduled launch time for the given Cron job. This important state change (recall it is done synchronously) ensures the Cron job schedules within the system are consistent. We keep track of all open launches, that is, launches where we have been notified of their beginning, but not of their end.

If a master replica dies or otherwise malfunctions (for example, gets partitioned away from the other replicas on the network), a slave should be elected as a new master. This election process must happen in less than one minute to avoid the risk of missing (or unreasonably delaying) a Cron job launch. Once a master is elected, all the open launches (that is, partial failures) must be concluded. This can be a complicated process, imposing additional requirements on both the Cron system and the datacenter infrastructure, and it deserves more detailed explanation.

Resolving partial failures. As mentioned earlier, the interaction between the master replica and the datacenter scheduler can fail in between sending multiple RPCs that describe a single logical Cron job launch, and we should be able to handle this condition as well.

Recall that every Cron job launch has two synchronization points: when we are about to perform the launch and when we have finished it. This allows us to delimit the launch. Even if the launch consists of a single RPC, how do we know whether the RPC was actually sent? We know the scheduled launch started, but we were not notified of its completion before the master replica died.

To achieve this condition, all operations on external systems, which we may need to continue upon re-election, either have to be idempotent (that is, we can safely do them again), or we must be able to look up their state and see whether they completed or not, unambiguously.

These conditions impose significant constraints, and may be difficult to implement, but they are fundamental to the accurate operation of a Cron service in a distributed environment that could suffer partial failures. Not handling this appropriately can lead to missed launches or a double launch of the same Cron job.

Most infrastructure that launches logical jobs in datacenters (for example, Mesos) provides naming for those jobs, making it possible to look up their state, stop them, or perform other maintenance. A reasonable solution to the idempotency problem is to construct these names ahead of time—without causing any mutating operations on the datacenter scheduler—and distribute them to all replicas of the Cron service. If the Cron service master dies during launch, the new master simply looks up the states of all the precomputed names and launches the missing jobs.

Recall that we track the scheduled launch time when keeping the internal state between the replicas. Similarly, we need to disambiguate our interaction with the datacenter scheduler, also by using the scheduled launch time. For example, consider a short-lived but frequently run Cron job. The Cron job was launched, but before this was communicated to all replicas, the master crashed and an unusually long failover—long enough that the Cron job finished successfully—took place. The new master looks up the state of the job, observes it finished, and attempts to launch it. If the time had been included, the master would have known the job on the datacenter scheduler was the result of this particular Cron job launch, and the double launch would not have happened.

The actual implementation has a more complicated system for state lookup, driven by the implementation details of the underlying infrastructure. However, the preceding description covers the implementation-independent requirements of any such system. Depending on the available infrastructure, you may also need to consider the tradeoff between risking a double launch and risking skipping a launch.

Storing the State

Using Paxos to achieve consensus is only one part of the problem of how to handle the state. Paxos is essentially a continuous log of state changes, appended to synchronously with the state changes. This has two implications: first, the log needs to be compacted, to prevent it from growing infinitely; second, the log itself must be stored somewhere.

In order to prevent infinite growth, we simply take a snapshot of the current state, so we can reconstruct the state without needing to replay all state change log entries leading to it. For example, if our state changes stored in logs are "Increment a counter by 1," then after a thousand iterations, we have a thousand log entries that can be easily changed to a snapshot of "Set counter to 1,000."

In case of loss of logs, we would only lose the state since the last snapshot. Snapshots are in fact our most critical state—if we lose our snapshots, we essentially are starting from zero because we have lost our internal state. Losing

logs, on the other hand, just causes a *bounded* loss of state and sends the Cron system back in time to when the latest snapshot was taken.

We have two main options for storing our data: externally in a generally available distributed storage, or in a system for storing the small volume of state as part of the Cron service itself. When designing the system, we opted for a bit of both.

We store the Paxos logs on local disk of the machine where Cron service replicas are scheduled. Having three replicas in default operation implies we have three copies of the logs. We store the snapshots on local disk as well. However, since they are critical, we also back them up onto a distributed file system, thus protecting against failures affecting all three machines.

We are not storing logs on our distributed file system, as we consciously decided that losing logs, representing a small amount of the most recent state changes, is an acceptable risk. Storing logs on a distributed file system can come with a substantial performance penalty caused by frequent small writes. The simultaneous loss of all three machines is unlikely, but in case it happens, we will automatically restore from the snapshot and lose only a small amount of logs since the last snapshot, which we perform at configurable intervals. Of course, these trade-offs may be different depending on the details of the infrastructure, as well as the requirements for a given Cron system.

In addition to the logs and snapshots stored on the local disk and snapshot backups on the distributed file system, a freshly started replica can fetch the state snapshot and all logs from an already running replica over the network. This makes replica startup independent of any state on the local machine, and makes rescheduling of a replica to a different machine upon restart (or machine death) essentially a non-issue for the reliability of the service.

Running a Large Cron

There are other, smaller, but no less interesting, implications of running a large Cron deployment. A traditional Cron is small: it probably contains at most tens of Cron jobs. However, if you run a Cron service for thousands of machines in a datacenter, your usage will match that, and you will run into consequent problems. A substantial problem is the distributed system's classic problem of the thundering herd, where the Cron service (based on user configuration) can cause substantial spikes in datacenter usage. When most people think of a daily Cron job, they immediately think of running it at midnight, and that is how they configure their Cron jobs. This works just fine if the Cron job launches on the same machine, but what if your job can spawn a MapReduce with thousands of workers? And what if 30 different teams decide to run daily Cron jobs like this in the same datacenter? To solve this problem, we introduced an extension to the crontab format.

In the ordinary crontab, users specify the minute, hour, day of the month (or week), and month when the Cron job should launch, or asterisk to mean every value. Running at midnight, daily, would then have crontab specification of "0 0 * * *" (that is, zeroth minute, zeroth hour, every day of the month, every month, and every day of the week). We have introduced the use of the question mark, which means any value is acceptable, and the Cron system is given the freedom to *choose* the value. This value is chosen by hashing the Cron job configuration over the given time range (for example, 0...23 for hour), thus distributing those launches more evenly.

Despite this change, the load caused by the Cron jobs is still very spiky. The graph in Figure 3 illustrates the aggregate number of launches of Cron jobs at Google, globally. This highlights the frequent spikes, which are often caused by jobs that must be launched at a specific time, for example, due to temporal dependency on external events.

Summary

A Cron service has been a fundamental feature in Unix systems for decades. The industry move toward large distributed systems, where a datacenter may be the smallest unit of hardware, requires changes in large portions of the stack, and Cron is no exception. A careful look at the required properties of a Cron service and the requirements of Cron jobs drives our new design.

We have discussed the new constraints and a possible design of a Cron service in a distributed system environment, based on the Google solution. The solution requires strong consistency guarantees in the distributed environment. The core of the distributed Cron implementation is therefore Paxos, a common algorithm for reaching consensus in an unreliable environment. The use of Paxos and correct analysis of new failure modes of Cron jobs in a large-scale, distributed environment allowed us to build a robust Cron service that is heavily used at Google.

Acknowledgments. Thanks to Dermot Duffy and Gabriel Krabbe (designers of the foundation of the system described in this article), Derek Jackson, Chris Jones, Laura Nolan, Niall Murphy, and all other contributors and reviewers for their valuable input.

Related articles on queue.acm.org

DNS Complexity
Paul Vixie
http://queue.acm.org/detail.cfm?id=1242499

Swamped by Automation
George Neville-Neil
http://queue.acm.org/detail.cfm?id=2440137

Principles of Robust Timing over the Internet
Julien Ridoux and Darryl Veitch
http://queue.acm.org/detail.cfm?id=1773943

References

1. Burrows, M. The Chubby lock service for loosely-coupled distributed systems. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*; (2006), 335–350; http://research.google.com/archive/chubby-osdi06.pdf
2. Corbett, J.C. et al. Spanner: Google's globally distributed database. In *Proceedings of the 10th Symposium on Operating System Design and Implementation* (2012); http://research.google.com/archive/spanner-osdi2012.pdf
3. Docker; https://www.docker.com/
4. Junqueira, F.P., Reed, B.C. and Serafini, M. Zab: High-performance broadcast for primary-backup systems. In *Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems & Networks* (2011), 245-256. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5958223&tag=1
5. Lamport, L. Paxos made simple. *ACM SIGACT News 32*, 4 (2001), 18–25; http://research.microsoft.com/en-us/um/people/lamport/pubs/pubs.html#paxos-simple
6. Lamport, L. Fast Paxos. *Distributed Computing 19*, 2 (2006), 79–103; http://research.microsoft.com/pubs/64624/tr-2005-112.pdf
7. Ongaro, D. and Ousterhout, J. In search of an understandable consensus algorithm (2014, extended version); https://ramcloud.stanford.edu/raft.pdf

Štěpán Davidovič is a site reliability engineer at Google on the Ads Serving SRE team charged with the reliability of the AdSense product. He also works on a wide range of shared infrastructure projects at Google.

Kavita Guliani is a technical writer for technical infrastructure and site reliability engineers at Google Mountain View. Previous employers include Symantec, Cisco, and Lam Research Corporation.

Copyright held by authors.
Publication rights licensed to ACM. $15.00.

DOI:10.1145/2666680

Even without it, some ingenious methods can be developed to help verify users' social media behavioral patterns.

BY REZA ZAFARANI AND HUAN LIU

Evaluation Without Ground Truth in Social Media Research

WITH THE RISE of social media, user-generated content is available at an unprecedented scale. These massive collections of user-generated content can help researchers better understand the online behavior of billions of individuals. This data also enables novel scientific research in social sciences, anthropology, psychology, and economics at scale.

Scientific research demands reproducible and independently verifiable findings. In social media research, scientific findings can be in the form of behavioral patterns, as in "Individuals commented on this Facebook post due to its quality." To validate these patterns, researchers can survey the individuals exhibiting a pattern to verify if it truly captured their intentions. In data mining terms, such validation is known as "evaluation with ground truth."[19] However, social media users are scattered around the world. Without face-to-face access to individuals on social media, is it even possible to perform an evaluation in social media research? That is, how can researchers verify the user behavioral patterns found are indeed the "true patterns" of these individuals?

This problem is known in the data mining and machine learning literature as "lack of ground truth," or lack of a gold standard that can help validate a scientific hypothesis. With so many opportunities for novel methods and findings on social media and limited ground truth, there is a pressing need for evaluation without ground truth in social media research. To address this problem systematically, researchers must consider the types of questions asked in social media research.

In social media research, researchers are often interested in when and where certain user activity is likely to take place and, if possible, why it takes place; for instance, a researcher might be interested in when individuals are more likely to tweet or search for restaurant reviews, at, say, a restaurant or home. Researchers also seek answers to "why" questions on social media: Why are users abandoning site A for site B?, and Why do users like Twitter despite the limit on the number of characters in a tweet? It is through such questions researchers look for inklings of causal-

» key insights

- With limited face-to-face access to social media users scattered around the globe, there is limited ground truth to help determine whether users' behavioral patterns obtained in social media research are indeed their true patterns.
- Researchers often focus on evaluating methods that predict when or where things happen on social media, why they happen, or the accuracy of the predictions.
- Subtle techniques can be borrowed from multiple disciplines or tweaked to help validate scientific discoveries on social media without ground truth.



IMAGE BY DONSCARPO MICROSTOCK

ity in social media. They are also interested in how an algorithm or incentive mechanism will work prior to its public release on social media. Companies often face similar questions when evaluating their methods on social media, from assessing the effectiveness of a new friend-recommendation algorithm to predicting the success of a new rewards program prior to release. Relevant recommendations and appropriate incentives can help increase user engagement and retention rates and, ultimately, sustain or increase profit.

Consider predicting when or where a particular user activity is going to happen. Unlike Apollo, the Olympian deity with knowledge of the future, humans find it a challenge to design methods able to predict the future. To answer, researchers design data-mining techniques that predict the most likely place or time an activity will happen in social media. The challenges introduced by humans' lack of knowledge about the future are further compounded by yearning to understand why things happen on social media. Without surveying users on social media, the gap between personal understanding and reality cannot be gauged.

Here, we discuss three types of evaluation for social media research that stand up to scientific scrutiny:

Spatiotemporal. In response to validating our discoveries on when or where things are going to happen, or spatiotemporal predictions, in social media;

Causality. Evaluating our hypotheses on causality, or why things are happening in social media; and

Outcome. Outcome-evaluation techniques assess how well a computational method (such as an algorithm, application, or incentive mechanism) predicts an outcome or finds patterns; outcome evaluation also helps determine how to improve the computational method to perform better.

Practitioners and researchers alike in various fields, including statistics, computer science, sociology, psychology, epidemiology, and ethology, have developed a range of methods social media researchers can borrow and tweak in their search for reproducible evaluation methods for social media research. We

present the most promising such methods, illustrating how they can be applied to social media research when ground truth is unavailable for evaluation.

Spatiotemporal Evaluation

Consider designing a method that predicts the most likely time users will check their email messages or the restaurant they will most likely choose for dinner using their check-ins, or personally reported locations, in social media. As the time or place predicted by the method occurs in the future, evaluation is a challenge. One straightforward heuristic that helps evaluate such spatiotemporal predictions is that individual behavior is periodic; for example, if an individual has checked email at a certain time for the past two days, it is then likely the same pattern will be observed in the same individual today (see Figure 1). The periodicity of human behavior simplifies evaluation for spatiotemporal predictions in social media.

Spatiotemporal predictions can be performed using a machine-learning method. Researchers collect historical data about an individual for a period of time. Following the common practice in machine learning, the data collected in this period is partitioned into two consecutive periods of training and testing; for example, for 10 days of data, the first seven days is used for training and the next three days for testing. The machine-learning method is provided with the training data, and its outcome is evaluated on the testing data. If the method performs well on the testing data, the researcher can safely assume that, due to the periodical nature of human behavior, it will perform well in the near future. Note we assume human behavioral patterns are consistent over time, which is often not true for long periods. The researcher must therefore train the machine-learning method repeatedly over time to ensure the technique consistently performs well.

What if a spatiotemporal prediction is for some behavior for which the researcher cannot guarantee recurrence, or periodicity, over time? One approach is crowdsourcing, using the wisdom of the crowd. In it, researchers ask opinions of multiple experts on a real-world problem (such as When will some phenomenon happen?) and choose the solution the majority agree on. Online labor markets, including Amazon's Mechanical Turk (http://www.mturk.com/), can provide a low-barrier entry to hiring individuals for performing crowdsourced evaluations on social media research.[3,15]

Ensemble methods[6] are a counterpart of crowdsourcing in machine learning; for instance, consider predicting whether or not someone will tweet. Researchers can employ statistically independent methods to predict whether or not an individual will tweet. These methods can be combined by considering the prediction to be what the majority of methods predict will happen, or that the user will or will not tweet. Because methods are independent, their predictions should be independent of one another. If more independent methods agree on a solution, the solution is likely a valid solution to the problem.[6,19,22]

Causality Evaluation

Consider how to determine the cause of a sudden increase in network traffic on a social media site. The researcher can speculate that some breaking news on the site is attracting users, resulting in a sudden increase in traffic. To determine the true cause of the traffic, one approach is to identify the counterfactual, or, in this case, what would have happened if the breaking news had not been released. That is, are there any other possible explanations (such as a malicious denial-of-service attack on the site) that could explain the heavy traffic? If no other explanation can be identified, then the researcher can safely say the breaking news is causing the traffic.

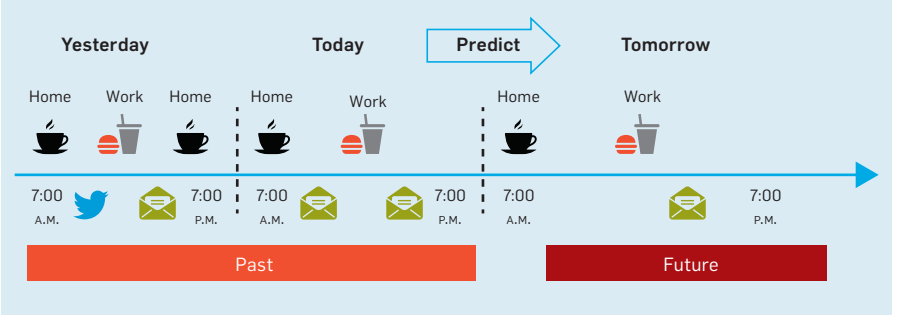
However, since determining the counterfactual requires knowing all other explanations, the researcher can resort to a controlled experiment if investigating the counterfactual is a challenge.

Controlled experiments. In a controlled experiment, users are randomly assigned to two groups: control and treatment. A treatment is administered to the treatment group, while the control group receives no treatment. If the treatment results in a significant outcome in the treatment group, then the researcher can safely say the treatment is causing the outcome. In our network-traffic example, the researcher releases the breaking news as a treatment to a random set of users (treatment group). The researcher then compares the level of traffic in this set to that of another random set of users for whom the breaking news is not shown (control group). If the traffic is significantly heavier in the treatment group, the researcher can conclude the breaking news is causing the traffic.

Note in a controlled experiment, the counterfactual is approximated without considering all other possible explanations. However, when taking random populations of users, a researcher is taking samples of all other explanations and comparing them to the treatment group that is shown the breaking news. To improve the confidence of controlled experiments, Sir Ronald A. Fisher proposed "randomized experiments" in which the researcher takes many random treatment populations and observes if the heavy traffic is observed in more than $(1-p)\%$ of the treatment groups; p denotes the confidence level, and a value of $p = 0.05$ is often considered in practice.

Controlled experiments are shown to be highly effective in social media re-

Figure 1. Spatiotemporal evaluation without ground truth exploits the periodicity of human behavior.



search. As an example, La Fond et al.[14] demonstrated how controlled experiments can be used to determine if influence is causing users to change behavior (such as by adding new hobbies due to a friend's influence). In these experiments, they generated control groups by randomizing user attributes (such as interests) over time. They assumed if influence exists, the influencer should become more similar to the influencee over time and this increase in similarity should be greater among influenced users than among randomly generated control groups.

Conducting controlled experiments is not always straightforward and can be financially and computationally expensive, and, at times, even impossible. So, a researcher must seek an alternative to find the counterfactual. A "natural experiment" is one such alternative.

Natural experiments. Nature often provides researchers and ordinary people alike a randomized experiment with no extra effort; for instance, consider a city in which a researcher is analyzing whether an increase in number of police officers will reduce the rate of street crime. To perform controlled experiments, the researcher must randomly select some neighborhoods, observe the change in crime rate after officers are assigned to them, and compare this change to those of other neighborhoods where officers are not assigned. However, the researcher cannot afford to perform randomized controlled experiments because deploying police officers is costly and experiments are time consuming. Klick et al.[9] said in cities (such as Washington, D.C.), this issue is easily addressed. Washington D.C. has a terrorist alert system due to the natural terrorist threat on the capital. The city responds to high-alert days by increasing the number of police officers in specific parts of the city. Assuming terrorist attacks are uncorrelated to crime in the streets, this existing plan to improve security provides a natural randomized experiment for measuring how numbers of officers affect crime. A researcher must observe only how crime rates change in the areas where officers are assigned due to terrorist attacks. In general, to perform a natural experiment, a researcher needs external knowledge about the phenomenon being studied. In our example, this external knowledge is the existence of the terrorist-alert system.

In social media research, researchers are often interested in when and where certain user activity is likely to take place and, if possible, why it takes place.

Natural experiments are also effective in social media research. As in controlled experiments, natural experiments can help identify influence in social media. Influence is known to be temporal; that is, the influential individuals take an action or adopt a product before their influencees do. Anagnostopoulos et al.[2] proposed to obtain randomized control groups by shuffling the times users are influenced in social media; for example, individuals who influence others to buy books must have read those books before the others, so by shuffling the times books are recommended and keeping the times when books were purchased intact, researchers can create randomized control groups. Their approach, known as the "shuffle test," identifies influence by measuring it on the original social media site and comparing it to its value on these randomized control groups. Likewise, Christakis et al.[5] showed that since influence is directional, or the influencer influences the influencee, the researcher can create randomized control groups by reversing the direction of interactions in social media; for instance, assume a researcher is measuring the influence a popular music star has over his or her fans. The researcher can observe and count the number of fans joining some site after the star joins the site. To compute influence for a randomized control, the researcher can reverse the influence direction by counting the number of fans joining the site before the star does.

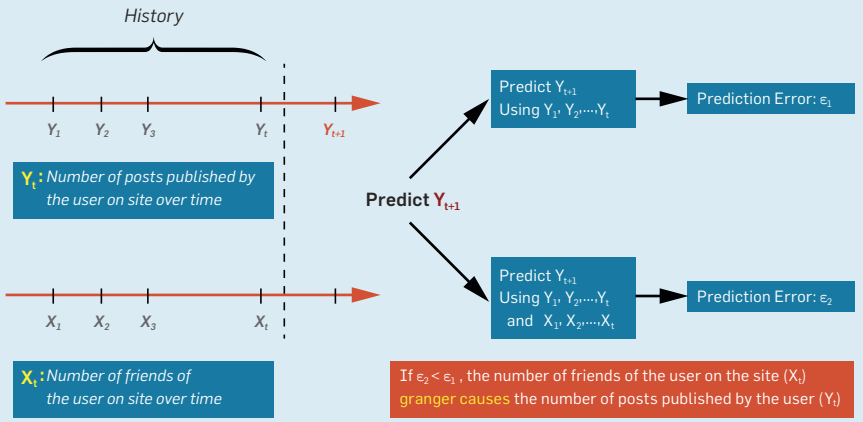
Natural experiments are difficult to perform in some cases, as they require searching for subtle ways to construct the control group. In general, a researcher might not be able to perform a natural experiment or have the luxury of randomized experiments. In this scenario, the researcher can resort to "nonequivalent control."

Nonequivalent control. In nonequivalent control, the control group is not selected randomly but such that the control group is similar to a randomized group. Consider user migrations across social media sites. Assume a researcher would like to determine if having no friends on one site will cause

Methods for determining true causality.

Method	Control Group	Treatment Group
Controlled Experiments	One or many manually randomized control groups	One or many manually treated treatment groups
Natural Experiments	Naturally random groups (randomization unnecessary)	Natural treatment groups (no treatment required)
Nonequivalent Control	One or many manually pseudo-randomized control groups	One or any manually treated treatment groups

Figure 2. Granger causality example.



a user to migrate to another site. The researcher can collect a random set of users across multiple sites; in it, users with no friends represent the treatment group. To create control groups, the researcher needs random subsets of users with friends. Instead, the researcher can construct similar control groups by taking random samples from the collected users who have friends. For both control and treatment groups, the researcher can observe user activity over time and determine if users migrate. If migrations are more significantly observed in treatment groups than in control groups, the researcher can safely say having no friends is likely to cause migrations. Kumar et al.[12] employed a similar approach to understand user migration patterns in social media.

All methods discussed thus far aim to determine "true" causality (see the table here). However, at times, determining "true" causality is impossible, and a researcher can determine only "pseudo-causality" by employing causality-detection techniques.

Causality detection. Assume the researcher would like to determine if the number of friends X_t a specific user has on a site at time t causes the number of posts Y_t the user publishes at that time. Causality-detection methods help validate this hypothesis by finding the relation between the two temporal variables $X = \{X_1, X_2, ..., X_t, X_{t+1}, ...\}$ and $Y = \{Y_1, Y_2, ..., Y_t, Y_{t+1}, ...\}$. A well-known causality-detection technique is "Granger causality,"[7,22] first introduced by Clive W.J. Granger, a Nobel laureate in economics.

Given two variables $X = \{X_1, X_2, ..., X_t, X_{t+1}, ...\}$ and $Y = \{Y_1, Y_2, ..., Y_t, Y_{t+1}, ...\}$, variable "$X$ Granger causes variable Y" when historical values of X help better predict Y than just using the historical values of Y (see Figure 2).

Consider a linear-regression model for predicting Y. The researcher can predict Y_{t+1} by using either $Y_1, ..., Y_t$ or a combination of $X_1, ..., X_t$ and $Y_1, ..., Y_t$,

$$Y_{t+1} = \sum_{i=1}^{t} a_i Y_i + \epsilon_1, \quad (1)$$

$$Y_{t+1} = \sum_{i=1}^{t} a_i Y_i + \sum_{i=1}^{t} b_i X_i + \epsilon_2, \quad (2)$$

where ϵ_1 and ϵ_2 are the regression model errors. Here, $\epsilon_2 < \epsilon_1$ indicates that using X helps reduce the error. In this case, X Granger causes Y.

Note Granger causality is "pseudo-causality," not equivalent to the "true" causality; for instance, a variable Z that causes both X and Y can result in X Granger causing Y. In this case, Granger causality is not equivalent to true causality.

Outcome Evaluation

Consider evaluating a method that estimates the "virality" of rumors in a social network without ground truth. The researcher expects the method to perform well for some user population, though the size of this population is often unknown. The researcher thus needs to take three independent steps:

Estimate magnitude. Estimate the size of the user population to which the rumor can spread. It is imperative to know the size of the population that has the opportunity to observe the rumor, as a rumor that is viral in a small population may not be as viral in a larger population;

Estimate sample accuracy. Given the population of users that has the opportunity to observe the rumor, the researcher often has limited access or computational power to analyze the whole population and must thus sample the population. This sample must accurately represent the general population. Unfortunately, random sampling is not always possible in the restricted ecosystems of social media; and

Estimate outcome. Over a sample, a researcher must determine how well the virality of rumors is estimated, despite lacking ground truth.

Figure 3 outlines these steps for the method that estimates rumor virality in a social network.

Magnitude estimation. For estimating the size of a population without ground truth, a researcher can resort to similar techniques designed in anthropology or ethology. In anthropology, the network scale-up method, or NSUM, introduced by Bernard et al.[4] is designed for magnitude estimation. NSUM was first introduced after the devastating 1985 earthquake in Mexico City to predict, after the earthquake had struck, the number of individuals likely to have perished during the earthquake. To estimate the number of

people likely to have perished, the researcher can ask a small set of individuals if they personally know someone who perished during the earthquake. By surveying a small set of individuals, the researcher can estimate p, the probability of individuals personally knowing someone who perished during the earthquake. Let n denote the size of the city, s the size of the subpopulation that perished, n_i^s the number of perished people personally known by individual i, and f_i the size of the population individual i knows personally (number of friends of individual i). NSUM predicts s as

$$s = n \times p = n \times \frac{\sum_i n_i^s}{\sum_i f_i}. \quad (3)$$

NSUM and variants[13] have been used successfully in social media research; for instance, researchers can estimate the number of users from a specific country in Facebook (s), using the number of users on Facebook (n), number of friends users have on Facebook (f_i), and number of friends they have from that country (n_i^s).

A similar approach, called "mark and recapture," is often used to estimate the population of animals, taking place in two phases: In the first, the researcher captures a set of animals and marks them; in the second, after some preset time, the researcher captures another set of animals and observes how many were recaptured. Let n denote the size of the animal population being estimated, m the total population marked in the first phase, c the population captured in the second phase, and r the population that was marked in phase one and recaptured in phase two. The mark-and-recapture technique then estimates n as

$$n = \frac{m \times c}{r}. \quad (4)$$

As with NSUM, this approach can be used to estimate population sizes in social media, as shown by Papagelis et al.[18]

Sample accuracy estimation. In social media, researchers often sample users or content from a site, despite not knowing how representative the samples actually are. Determining effective sample size is well studied in statistics and in survey sampling[8] when the sampling technique is known. But when the sampling technique is unknown, the researcher can generate a small random sample and compare it to the sample obtained through the unknown technique. This way the researcher can empirically determine how close the unknown samples are to random samples; for instance, Morstatter et al.[16] in 2013 estimated how representative tweet samples collected from Twitter's API are by comparing them to the random samples collected from all the tweets on Twitter during the same period. By comparing samples through statistical measures, they concluded that small samples from Twitter's API might, in some cases, not be representative of the general tweets observed in Twitter.

Outcome estimation. When a representative sample is available, evaluating the performance of a computational method on it without ground truth can be a challenge. Depending on whether user feedback or controlled experiments are available from the social media site, different techniques can be employed to estimate outcome.

Unavailable feedback. When feedback is not available, the researcher can use external sources or computational methods to perform the evaluation; for instance, consider a method that identifies influential users on a site. To verify they are correctly identified, the researcher can seek validating evidence on other sites; for instance, Agarwal et al.[1] verified influential bloggers identified by their algorithm on The Unofficial Apple Weblog (TUAW) using information available on the Digg news-aggregator site. They assumed influential bloggers on TUAW should be praised and cited more often on sites like Digg. As another example of unavailable feedback, consider a method that identifies user communities on a social network. As such communities are often unknown, the researcher can evaluate identified communities through methods that quantify the intuition that a community corresponds to a densely interact-

Figure 3. An example of outcome evaluation without ground truth.

Here, we are evaluating a method that estimates the "virality" of rumors in a social network without ground truth. Green nodes are users who have not received the rumor yet. Blue nodes are users who have received the rumor and have adopted (such as by posting) it. Yellow nodes are users who have received the rumor but rejected it. As there is no ground truth, the method that estimates the virality of rumors lacks access to node colors.

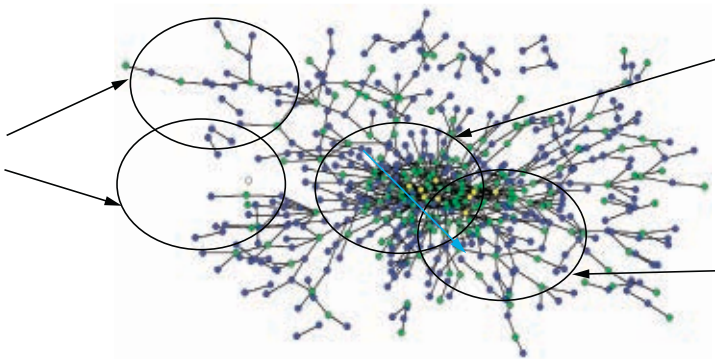
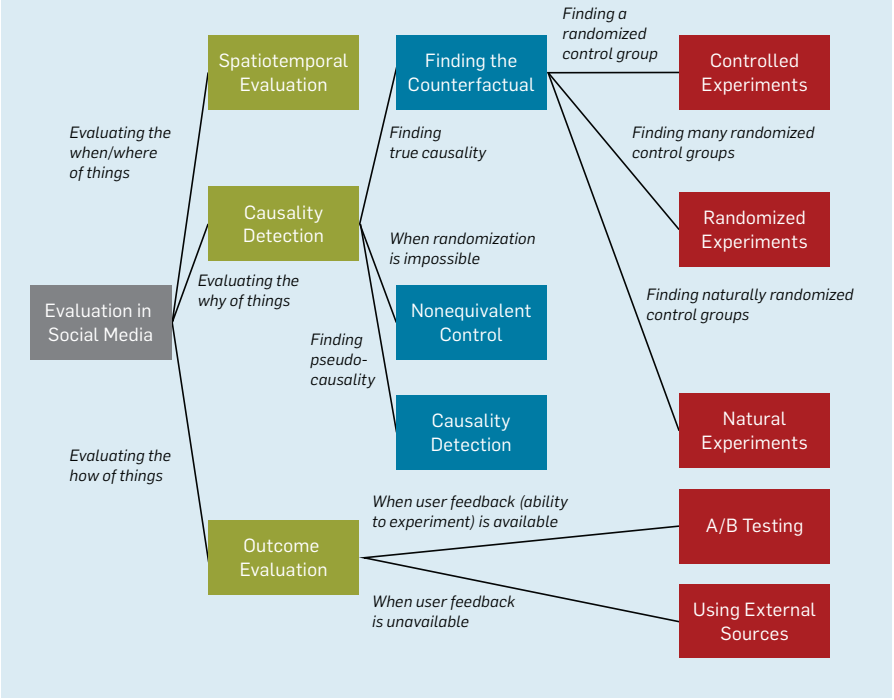


Figure 4. Map for evaluation without ground truth in social media.



ing set of users.[20] One such method is "modularity,"[17] which is often used for evaluating community-detection algorithms in social media when ground truth is unavailable.[21]

Available feedback. When feedback is available, the researcher can perform controlled experiments to evaluate. In the industry, these experiments are often referred to as A/B testing;[10,11] for example, a social media company tests a new feature for its site by dividing users on the site randomly into two groups—A and B. Group A is shown the feature, and for its users, some quantity (such as number of visits to the site) is measured. Group B is shown the same site but without the feature, and its number of visits to the site is measured as well. If the difference between the number of visits in group A is significantly larger than the number of visits in group B, then the feature is beneficial to the site and can be expected to increase the traffic.

Conclusion

Evaluation of social media means new challenges. One is the lack of ground truth. However, proven scientific methods can be borrowed and tweaked for evaluating social media-research findings.

Here, we have discussed three general categories of evaluation on social media: spatiotemporal, causality, and outcome. Spatiotemporal evaluation can be performed knowing that humans exhibit periodical spatiotemporal behavior. Causality evaluation can be performed by investigating the counterfactual, performing controlled, randomized, or natural experiments, finding nonequivalent controls, or employing causality detection techniques. When evaluating outcome, three tasks must be performed: estimating magnitude, estimating sample accuracy, and estimating outcome (see Figure 4). Learning and applying methods from statistics, anthropology, and ethology can help researchers perform these tasks efficiently. In addition, these methods help advance research in social media and inspire development of novel evaluation methods for new research needs. C

References

1. Agarwal, N., Liu, H., Tang, L., and Yu, P.S. Identifying the influential bloggers in a community. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (Stanford, CA, Feb. 11–12). ACM Press, New York, 2008, 207–218.
2. Anagnostopoulos, A., Kumar, R., and Mahdian, M. Influence and correlation in social networks. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery in Databases* (Las Vegas, Aug. 24–28), ACM Press, New York, 2008, 7–15.
3. Barbier, G., Zafarani, R., Gao, H., Fung, G., and Liu, H. Maximizing benefits from crowdsourced data. *Computational and Mathematical Organization Theory 18*, 3 (Sept. 2012), 257–279.
4. Bernard, H.R., Johnsen, E.C., Killworth, P.D., and Robinson, S. Estimating the size of an average personal network and of an event subpopulation: Some empirical results. *Social Science Research 20*, 2 (June 1991), 109–121.
5. Christakis, N.A. and Fowler, J.H. The spread of obesity in a large social network over 32 years. *New England Journal of Medicine 357*, 4 (Nov. 2007), 370–379.
6. Dietterich, T.G. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems* (Cagliari, Italy, June 21–23), Springer, Berlin, 2000, 1–15.
7. Granger, C.W.J. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society 37*, 3 (July 1969), 424–438.
8. Kish, L. *Survey Sampling.* John Wiley & Sons, Inc., New York, 1965.
9. Klick, J. and Tabarrok, A. Using terror alert levels to estimate the effect of police on crime. *Journal of Law and Economics 48*, 1 (Apr. 2005), 267–279.
10. Kohavi, R., Henne, R.M., and Sommerfield, D. Practical guide to controlled experiments on the Web: Listen to your customers, not to the hippo. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery in Databases* (San Jose, CA, Aug. 12–15). ACM Press, New York, 2007, 959–967.
11. Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R.M. Controlled experiments on the Web: Survey and practical guide. *Data Mining and Knowledge Discovery 18*, 1 (Feb. 2009), 140–181.
12. Kumar, S., Zafarani, R., and Liu, H. Understanding user migration patterns in social media. In *Proceedings of the 25th Conference of the Association for the Advancement of Artificial Intelligence* (San Francisco, Aug. 7–11). AAAI, Palo Alto, CA, 2011, 1204–1209.
13. Kurant, M., Gjoka, M., Butts, C.T., and Markopoulou, A. Walking on a graph with a magnifying glass: Stratified sampling via weighted random walks. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems* (San Jose, CA, June 7–11). ACM Press, New York, 2011, 281–292.
14. La Fond, T. and Neville, J. Randomization tests for distinguishing social influence and homophily effects. In *Proceedings of the 19th WWW Conference* (Raleigh, NC, Apr. 26–30). ACM Press, New York, 2010, 601–610.
15. Mason, W. and Suri, S. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods 44*, 1 (Mar. 2012), 1–23.
16. Morstatter, F., Pfeffer, J., Liu, H., and Carley, K.M. Is the sample good enough? Comparing data from Twitter's streaming API with Twitter's firehose. In *Proceedings of the Seventh International Conference on Weblogs and Social Media* (Cambridge, MA, July 8–11). AAAI Press, Palo Alto, CA, 2013, 400–408.
17. Newman, M.E.J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences 103*, 23 (June 2006), 8577–8582.
18. Papagelis, M., Das, G., and Koudas, N. Sampling online social networks. *IEEE Transactions on Knowledge and Data Engineering 25*, 3 (Mar. 2013), 662–676.
19. Tan, P.N., Steinbach, M., and Kumar, V. *Introduction to Data Mining.* Pearson Addison-Wesley, Boston, MA, 2006.
20. Yang, J. and Leskovec, J. Defining and evaluating network communities based on ground truth. *Knowledge and Information Systems 42*, 1 (Jan. 2015), 181–213.
21. Yang, T., Chi, Y., Zhu, S., Gong, Y., and Jin, R. Detecting communities and their evolutions in dynamic social networks a Bayesian approach. *Machine Learning 82*, 2 (Feb. 2011), 157–189.
22. Zafarani, R., Abbasi, M.A., and Liu, H. *Social Media Mining: An Introduction.* Cambridge University Press, New York, 2014.

Reza Zafarani (reza@asu.edu) is a Ph.D. candidate in computer science and member of the Data Mining and Machine Learning Laboratory at Arizona State University, Tempe, AZ.

Huan Liu (huan.liu@asu.edu) is a full professor of computer science and engineering and director of the Data Mining and Machine Learning Laboratory at Arizona State University, Tempe, AZ.

© 2015 ACM 00010782/15/06 $15.00

DOI:10.1145/2663340

Formal executable models enable systematic evaluation of system designs prior to implementation and deployment.

BY KURT JENSEN AND LARS M. KRISTENSEN

Colored Petri Nets:

A Graphical Language for Formal Modeling and Validation of Concurrent Systems

THE VAST MAJORITY of IT systems today can be characterized as concurrent and distributed in that their operation inherently relies on communication, synchronization, and resource sharing between concurrently executing software components and applications. This development was accelerated first through the pervasive presence of the Internet as a communication infrastructure and more recently by cloud- and Web-based services, mobile applications, and multicore computing architectures.

» key insights

- **Concurrent and distributed systems consist of parallel processes that communicate, synchronize, and share resources; such systems are a challenge to design and implement correctly, as they exhibit complex nondeterministic behavior.**
- **Colored Petri Nets extend Petri nets with data types, functions, and modules to obtain a scalable formal language suited for modeling concurrency, synchronization, and data processing.**
- **CPN Tools is a widely used software tool supporting construction, simulation, and analysis of executable models of concurrent systems.**

Colored Petri Nets, or CPNs, combine Petri nets with a programming language to obtain a scalable modeling language for concurrent systems. Petri nets provide the formal foundation for modeling concurrency and synchronization; a programming language provides the primitives for modeling data manipulation and creating compact and parameterizable models. In this article, we offer an example-driven introduction to the core syntactical and semantical constructs of the CPN modeling language and survey how quantitative and qualitative behavioral properties of CPN models can be validated through simulation-based performance analysis

Petri Nets

A Petri net in its basic form is called a place/transition net, or PTN, and is a directed bi-partite graph with nodes consisting of places (drawn as ellipses) and transitions (drawn as rectangles). The state of a Petri net is called a marking and consists of a distribution of tokens (drawn as black dots) positioned on the places. The execution of a Petri net (also called the "token game") consists of occurrences of enabled transitions removing tokens from input places and adding tokens to output places, as described by integer arc weights, thereby changing the current state (marking) of the Petri net. An abundance of structural analysis methods (such as invariants and net reductions), as well as dynamic analysis methods (such as state spaces and coverability graphs), exist for Petri nets.[9]

and explicit state-space exploration. We also give an overview of CPN Tools that provide software-tool support for the practical use of CPNs and pointers to projects where CPN technology has been put to practical use in an industrial setting. As we proceed, we give a historical perspective on the research contributions that led to development of CPN technology.

History of Colored Petri Nets

Development of CPNs was initiated in the early 1980s when distributed systems were becoming a major paradigm for future computing systems. The goal of the CPN modeling language has been to develop a formally founded modeling language for concurrent systems that would make it possible to formally analyze and validate concurrent systems and that, from a modeling perspective, scale to industrial systems. A main motivation behind the research into CPNs (and many other formal modeling languages) was that the engineering of correct concurrent systems is a challenge due to their complex behavior that could result in subtle bugs if not designed with care. As concurrent systems become still more pervasive and critical to society, formal techniques for concurrent systems were (and still are) a highly relevant technology to support engineering of reliable concurrent systems. Many formal modeling languages for concurrent systems have been developed, including Statecharts,[10] Timed Automata,[1] and Promela.[11]

At their origin, CPNs build on Petri nets (see the first sidebar "Petri Nets") as introduced by Carl Adam Petri in his doctoral thesis published in 1962[21] as a formalism for concurrency and synchronization. Petri's introduction of Petri nets was far ahead of the time distributed systems would be invented and computers begin to have parallel processes. At that time, programs and processing were considered sequential and deterministic. It was thus extremely visionary of Petri to predict the importance of being able to understand and characterize the basic concepts of concurrency. In Petri nets, actions are (implicitly) concurrent unless (explicitly) synchronized; this is in contrast to many other modeling formalisms where concurrency must be introduced explicitly using parallel composition operators. A further advantage of Petri nets is they rely on few basic concepts yet are still able to model a range of communication and synchronization concepts and patterns.

In the decade following their introduction, Petri nets were widely accepted as one of the most well-founded theories to describe important behavioral concepts, including concurrency, conflict, nondeterminism, synchronization, and resource sharing. Petri nets were also used to model and analyze smaller concurrent systems. However, practical use soon revealed a serious shortcoming: Petri nets (in their basic form) do not scale to large systems unless the system is modeled at a very high level of abstraction. The primary reason for this lack of scalability is that basic Petri nets are not well suited for modeling systems in which data and manipulation of data play a crucial role. Furthermore, Petri nets did not provide concepts that would make it easy to scale models according to some system parameter (such as increase the number of servers in a modeled system) without having to make major changes to the model. This trade-off implied the use of Petri nets for practical modeling was declining. As a remedy, many researchers proposed ad hoc extensions to Petri nets. These extensions created a large variety of Petri net modeling languages. Many ad hoc extensions were not well defined and, even when they were, often had fundamental theoretical problems. Whenever a new ad hoc extension was introduced, all the basic concepts and analysis methods had to be redefined to apply for the extended Petri net language (with ad hoc extension). With the invention of the first (text-based) computer tools to support analysis of Petri net models, the situation became prohibitive. Whenever a new ad hoc extension was introduced (to handle a modeling shortcoming) all existing computer tools became void and could be used only after time-consuming and error-prone reprogramming. Hence, there was an urgent need to develop a class of Petri nets general enough to handle a large variety of different application areas without having to make ad hoc extensions.

The first successful step toward a common more-powerful class of Petri nets was taken by Hartmann Genrich and Kurt Lautenbach in 1979 with the introduction of predicate/transition, or PrT, nets.[8] Their work was inspired by earlier work on transition nets with "colored tokens" by Michael Schiffers and Horst Wedde[23] and transition nets with "complex conditions" by Robert M. Shapiro.[24] The basic idea behind PrT nets was to introduce a set of colored tokens that can be distinguished from one another, unlike the indistinguishable black tokens in basic Petri nets. It was thus possible to model different processes in a single subnet. PrT nets used arc expressions to define how transitions can occur in different ways (occurrence modes) depending on the colors of input and output tokens. The invention of colored distinguishable tokens in PrT nets was a significant step forward but still involved limitations. PrT nets had only one set of token colors, and all places had to use this set (or Cartesian products based on this set).

The second step toward a more general class of Petri nets was taken by Kurt Jensen in his Ph.D. thesis in 1980 with the introduction of the first kind of CPNs.[14] This Petri net model

allowed the modeler to use a number of different color sets, making it possible to represent data values in a more intuitive way instead of having to encode all data into a single shared set. It later turned out to be convenient to define the color sets by means of data types known from programming languages (such as products, records, lists, and enumerations). Use of data types had three implications: Token colors became structured (and hence much more powerful); type checking became possible (making it much easier to locate modeling errors); and color sets, arc expressions, and guards could be specified by the well-known and powerful syntax and semantics known from programming languages. Use of data types gave the modeler a convenient way to handle complex data and specify the often-complex interaction between data and system behavior.

A third step forward was taken by Huber et al. in 1990 with the introduction of hierarchical CPNs,[12] work heavily inspired by the hierarchy concepts in the Structured Analysis and Design Technique (SADT) developed by Marca and McGowan.[18] It was Shapiro of Huber et al. who proposed to port the SADT hierarchy concepts to CPNs. Introduction of hierarchical CPNs allowed the modeler to structure a large CPN model into a number of interacting and reusable modules in a way that is similar to how modules are used in programming languages. Development of hierarchical CPNs implied that Petri net models of large systems became much more tractable, as they could be split into modules of reasonable size, and the model could be viewed at different levels of abstraction.

The CPN Language

To give an informal introduction to the concepts of CPNs, we use a CPN model of a distributed two-phase commit transaction system; for more on the formal definition of CPNs, see the CPN lecture book by Jensen and Kristensen.[16] The CPN model of the two-phase commit system consists of four modules hierarchically organized into three levels; Figure 1 outlines the top-level module, which consists of two "substitution transitions" (drawn as rectangles with double-lined borders) representing the `Coordinator` and the `Workers` in the system. Each substitution transition has an associated "submodule" that models the detailed behavior of the coordinator and the workers, respectively. Substitution transitions constitute a purely syntactical structuring mechanism and do not occur (execute) when CPN models are executed.

The two substitution transitions are connected via directed "arcs" to the four places: `CanCommit`, `Votes`, `Decision`, and `Acknowledge` (drawn as ellipses). Places connected to substitution transitions are called "socket places" and are linked to "port places" in the associated submodules (to be discussed shortly). The coordinator and the workers interact by producing and consuming "tokens" on the places. These tokens carry data values; the type of tokens that may reside on a place is determined by the "type" of the place (written in text below the place). For historical reasons, the types of places are called "color sets"; Figure 2 lists the definitions of the color sets used for the four places in Figure 1. These color sets are defined using the CPN ML programming language, which is based on the functional language Standard ML.[19]

The symbolic constant `W` is used to specify the number of worker processes considered. The Worker color set is an indexed type consisting of the values `wrk(1),wrk(2) ... wrk(W)` used to model the identity of the worker processes. The color set `Vote` is an enumeration type containing the values `Yes` and `No` and used to model that a worker may vote yes or no to commit the transaction. The color set `WorkerxVote` is a product type containing pairs consisting of a worker and

Figure 1. The top-level module of the CPN model.

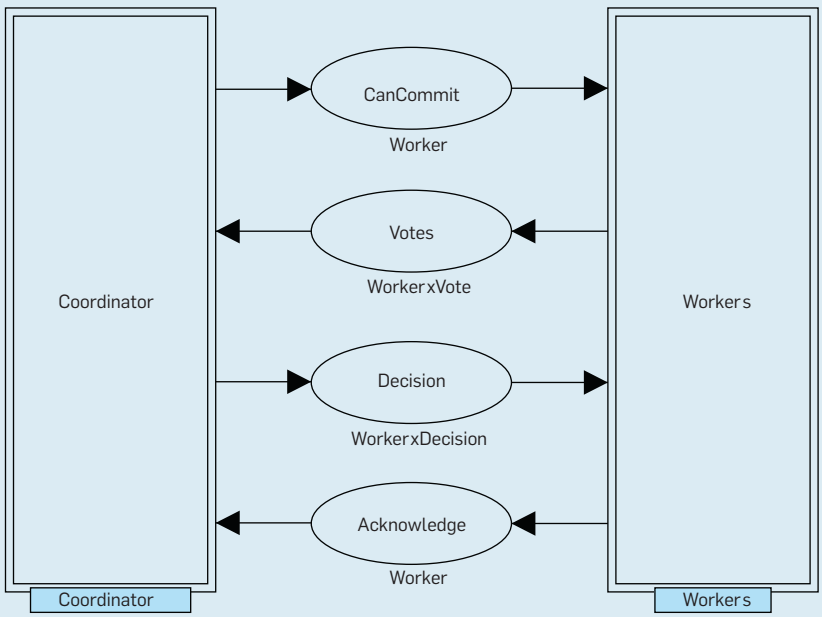


Figure 2. Definition of color sets used in Figure 1.

```
val W = 2;

colset Worker = index wrk with 1..W;
colset Vote = with Yes | No;
colset WorkerxVote = product Worker * Vote;

colset Decision = with Abort | Commit;
colset WorkerxDecision = product Worker * Decision;

var w : Worker;
var vote : Vote;
```

Figure 3. The `Coordinator` module.

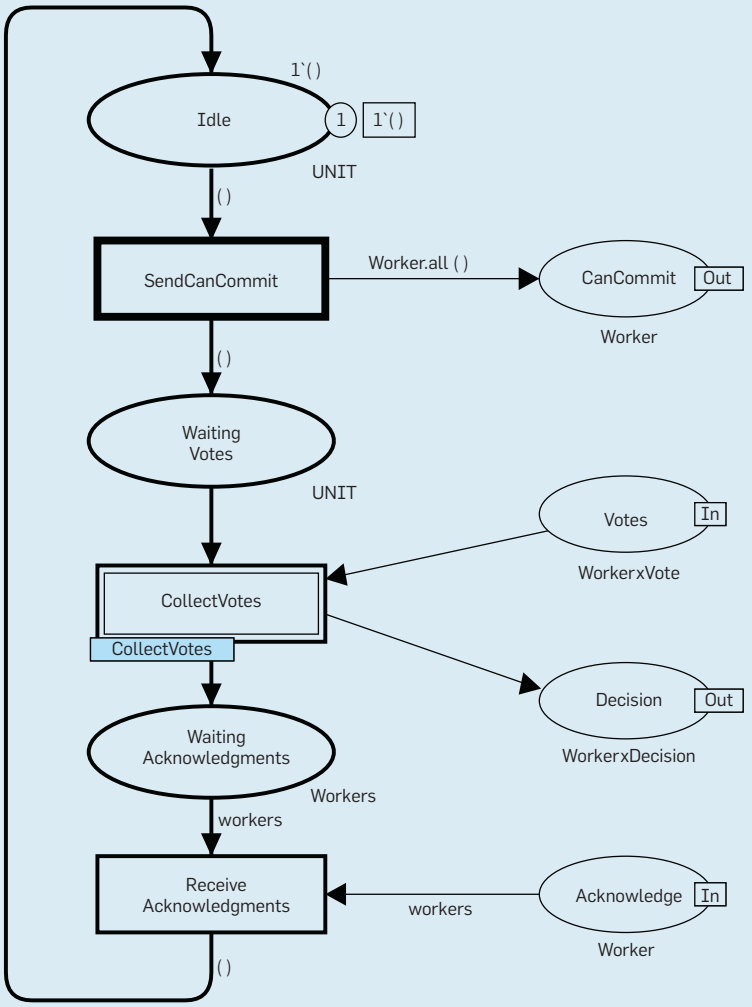
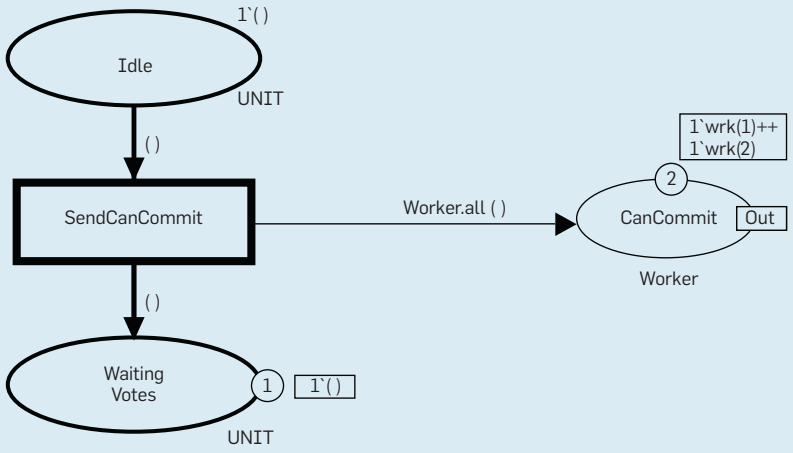


Figure 4. Current marking after `SendCanCommit`.



its vote. The color set `Decision` is an enumeration type used to model whether the coordinator decides to `Abort` or `Commit` the transaction; only if all workers vote yes will the transaction be committed. Note that in addition to the type constructors introduced here, CPN ML supports union, list, and record types. The variables `w` and `vote` declared in Figure 2 are discussed later.

The state of a CPN model is called a "marking" and consists of a distribution of "tokens" on the places of the model. Each place may hold a (possibly empty) "multi-set" of tokens with data values (colors) from the color set of the place. The "initial marking" of a place is specified above each place (omitted if the initial marking is the empty multi-set). For the places in Figure 1, all places are empty in the initial marking. Initially, the "current marking" of a CPN model equals the initial marking. When a CPN model is executed "occurrences" of "enabled" transitions consume and produce colored tokens on the places changing the current marking of the CPN model.

The `Coordinator` module (see Figure 3) is the submodule associated with the `Coordinator` substitution transition in Figure 1. The places `CanCommit`, `Votes`, `Decision`, and `Acknowledge` are "port places," as indicated by the rectangular `In` and `Out` tags positioned next to them. These places are linked to the identically named socket places in the top-level module in Figure 1 via a "port-socket association," implying any tokens added (removed) from a port place by transitions in the `Coordinator` module will also be added (removed) in the marking of the associated socket place in the top-level module.

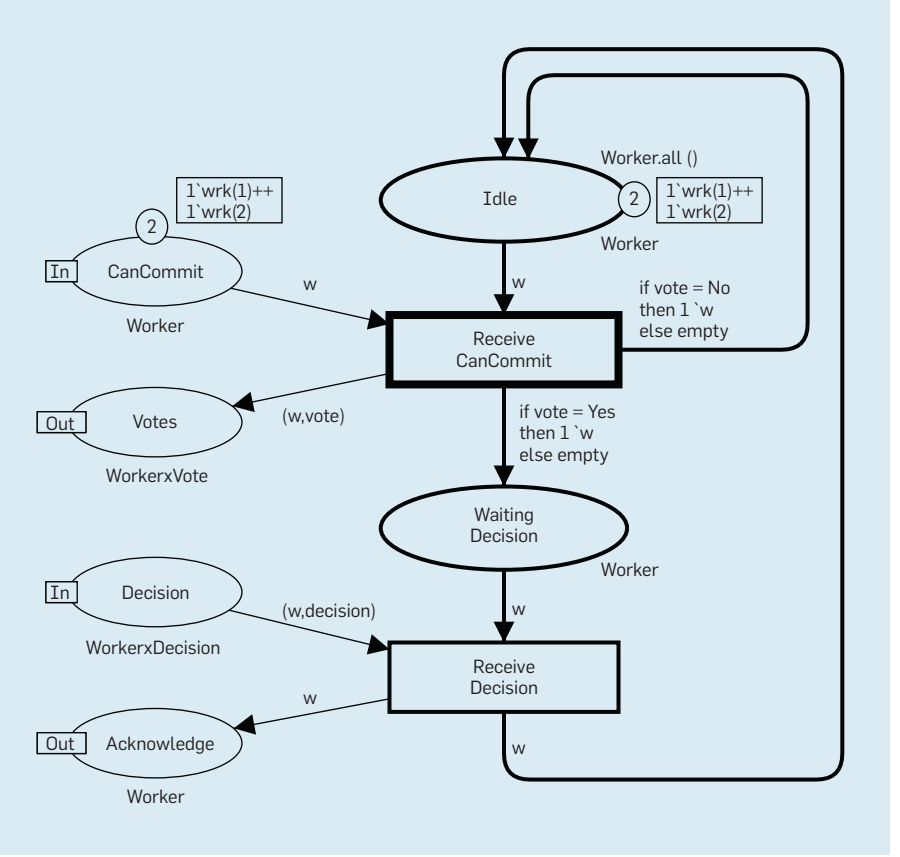
The places `Idle`, `WaitingVotes`, and `WaitingAcknowledgements` are used to model the states of the coordinator when executing the two-phase commit protocol. The places `Idle` and `WaitingVotes` have the color set `UNIT` containing just a single value `()`(denoted unit). Initially, the coordinator is in an idle state, as modeled by the initial marking of place `Idle`, which consists of a single token with the color unit. In CPN ML this is written as `1`()` specifying one

(1) occurrence of (`) the unit color (()). The number of tokens on a place in the current marking is indicated with a small circle positioned next to a place, and the details of the colors of the tokens are provided in an associated text box.

The transitions SendCanCommit, CollectVotes, and ReceiveAcknowledgements model the events/actions that cause the coordinator to change state. The coordinator will first send a can commit message (transition SendCanCommit) to each worker asking whether the workers can commit the transaction. It will then collect the votes from all workers and send a decision to the workers that voted yes indicating whether the transaction is to be committed or not (substitution transition CollectVotes). Finally, the coordinator will receive an acknowledgment from each worker that voted yes, confirming the worker received the decision (transition ReceiveAcknowledgements). Note CollectVotes is a substitution transition, meaning the details of how the coordinator collects votes are modeled by the associated CollectVotes submodule. This illustrates the mixed use of ordinary and substitution transitions within a module. We omit the details of CollectVotes in this article.

In the current marking in Figure 3, only the transition SendCanCommit is enabled, as indicated by the thick border of that transition. The requirements for a transition to be "enabled" are determined from the "arc expressions" associated with the incoming arcs of the transition. In this case, only a single incoming arc is from place Idle containing the expression (). This expression specifies that for SendCanCommit to be enabled, there must be at least one ()-token present on Idle. When the SendCanCommit transition occurs, it consumes a ()-token from place Idle and produces tokens on places connected to output arcs, as determined by "evaluating" the arc expressions on output arcs. In this case of the SendCanCommit transition, the expression () on the arc to WaitingVotes evaluates to a single ()-token. The expression Worker.all() on the arc to CanCommit is a call to the function Worker.all

Figure 5. The Workers module.



that takes a unit value (()) as parameter and returns all colors of the color set Worker. The use of Worker.all() as arc expression illustrates how complex calculations can be encapsulated within function calls. The calculations within functions may use complex data values and involve multi-sets of tokens such that several tokens can be added/removed by a single transition occurrence without introducing intermediate states.

Figure 4 outlines the marking of the surrounding places of transition SendCanCommit after the occurrence of SendCanCommit, showing the place CanCommit contains two tokens—one for each worker in the system—representing messages going to the two worker processes. The coordinator has now entered a state in which it is waiting to collect votes from the worker processes.

Figure 5 is the Workers module, which is the submodule of the Workers substitution transition in Figure 1. The places CanCommit, Votes, Decision, and Acknowledge constitute the port places of this module and are linked to the identically named socket places in Figure 1. The places Idle and WaitingDecision model the two states of worker processes. Each place has the color set Worker. A wrk(i)-token on, say, the place Idle reflects that the i-th worker is in state idle. This modeling approach makes it possible to model the states of all workers in a compact manner within a single module without having to have a place for each worker or a module instance for each worker. All workers are initially in the idle state, as represented by corresponding tokens on place Idle in the initial marking. The transition ReceiveCanCommit models the reception of can-commit messages from the coordinator and the sending of a vote. The transition ReceiveDecision models the reception of a decision message from the coordinator and the sending of an acknowledgment.

The current marking of place CanCommit in Figure 5 is 1`wrk(1) ++ 1`wrk(2) modeling a marking where the coordinator has sent a can-commit message to each worker. The arc expressions on the surrounding arcs of the ReceiveCanCommit

transition are more complex than the arc expressions of the SendCanCommit transition in the Coordinator module considered earlier in that they contain the "free variables" w and vote defined in Figure 2. This means in order to talk about the enabling and occurrence of transition ReceiveCanCommit, values must be bound (assigned) to w and vote in order to evaluate the input and output arc expressions. This is done by creating a "binding" that associates a value to each of the free variables occurring in the arc expressions of the transition. Bindings can be considered different "modes" in which a transition may occur. As w is of type Worker and vote is of type Decision, this yields the bindings listed in Figure 6, reflecting that each of the two workers may vote Yes or No to committing the transactions.

A binding of a transition is enabled if evaluating each input arc expression in the binding results in a multi-set of tokens, which is a subset of the multi-set of tokens present on the corresponding input place; as an example, consider the binding b_{1Y} for transition ReceiveCanCommit. Evaluating the input arc expression w on the input arc from Idle results in the multi-set containing a single token with the color wrk(1) contained in the multi-set of tokens present on place Idle in the marking in Figure 5, similarly for the input arc expression on the arc from place CanCommit. This means that binding b_{1Y} is enabled and may occur. In fact, all four bindings listed in Figure 6 are enabled in the marking in Figure 5.

The tokens produced on output places when a transition occurs in an enabled binding are determined by evaluating the output arc expressions of the transition in the given binding. Consider again the binding b_{1Y} for transition ReceiveCanCommit. The output arc expression (w,vote) evaluates to (wrk(1),Yes), and this token will be added to place Votes to inform the coordinator that worker one votes yes to committing the transaction. The arc expression on the arc from ReceiveCanCommit to WaitingDecision is an if-then-else expression that in the binding b_{1Y} evaluates to the multi-set 1`wrk(1) that will be added to the tokens on place WaitingDecision. The if-then-else expression on the arc from ReceiveCanCommit to Idle evaluates to the empty multi-set; hence, no tokens will be added to place Idle in this case. Figure 7 outlines the marking resulting from an occurrence of the b_{1Y} binding.

The occurrence of the binding b_{1N} representing that worker one votes no would have the effect of removing a wrk(1)-token from Idle, adding a (wrk(1),No)-token to Votes, adding no tokens to place WaitingDecision, and adding a wrk(1)-token to place Idle. This models the fact that if a worker votes no to committing the transaction, it then goes back to Idle; whereas if it votes yes, it will then go to WaitingDecision. As this is a distributed system, a worker cannot know the vote of another worker without exchanging messages.

All four bindings listed for transition ReceiveCanCommit are enabled in the marking shown in Figure 5. Enabled bindings may be "concurrently enabled" if each binding can get its required tokens from each input place independently of the other enabled bindings in the set. As an example, the bindings b_{1Y} and b_{2Y} are concurrently enabled since each binding can get its tokens from the input places without sharing with each other. Such access to tokens reflects that the workers are executing concurrently and may simultaneously send a vote back to the coordinator. In contrast, the bindings b_{1Y} and b_{1N} are not concurrently enabled. These two bindings are in "conflict" because they each need the single wrk(1)-token on Idle (and also the single wrk(1)-token on place CanCommit). The notion of concurrency and conflicts of bindings extends to bindings of different transitions. A concurrently enabled set of bindings can be executed in any interleaved order, and the resulting marking will be the same, independent of the interleaved execution considered, a property inherited from basic Petri nets.

So far, we have used relatively simple arc expressions. In general, arc expressions can be any expression that can be written in Standard ML as long as they have types that match the corresponding places. In particular, arc expressions may apply functions, including higher-order functions.

Figure 6. Bindings of transition ReceiveCanCommit.

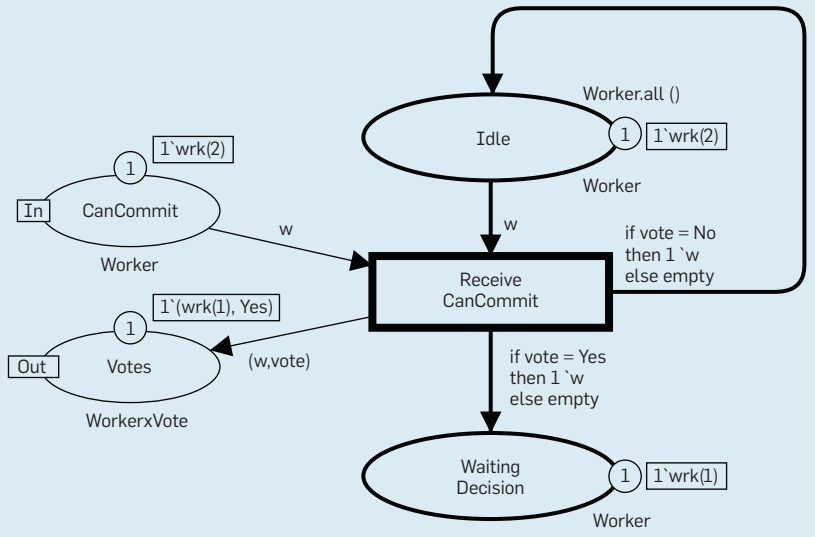
b_{1Y} = <w = wrk(1), vote = Yes>

b_{1N} = <w = wrk(1), vote = No>

b_{2Y} = <w = wrk(2), vote = Yes>

b_{2N} = <w = wrk(2), vote = No>

Figure 7. Current marking after ReceiveCanCommit.



Unfolding and Folding

By adding data types, a programming language, and modules, CPNs improve practical modeling power, making it feasible to create models of complex real systems. PTNs extended with inhibitor arcs allowing a test for zero tokens on a place can already simulate Turing machines; hence, the CPN extensions do not add expressive power, from a theoretical perspective. Moreover, any hierarchical CPN model can be unfolded to a nonhierarchical CPN model that in turn can be unfolded to a behaviorally equivalent PTN model. In the other direction, any PTN model, including one with inhibitor arcs, can be folded into a CPN model with a single module consisting of a single place and a single transition. In practice, such folding is not interesting, as the arc expressions will be extremely complex and not interpretable by a human. However, the fact that the unfolding and folding exist shows hierarchical CPNs have the same behavioral properties as basic Petri nets, and hence constitute a solid model for concurrency, conflict, synchronization, and resource sharing.

The unfolding of a hierarchical CPN to a nonhierarchical CPN consists of recursively replacing each substitution transition with its associated submodule such that associated port and socket places are merged into a single place. The unfolding of a nonhierarchical CPN to a PTN consists of unfolding each CPN place to a PTN place for each color in the color set of the CPN place and unfolding each CPN transition to a PTN transition for each possible binding of the CPN transition. To illustrate the unfolding, consider the subnet in Figure 4. To represent this CPN subnet as a PTN, the CPN place `CanCommit` must be unfolded to two PTN places corresponding to `wrk(1)` and `wrk(2)`. The places `Idle` and `WaitingVotes` do not need to be unfolded, as the unit color set contains only a single value. Figure 8 shows the equivalent PTN; arc expressions are replaced by arc weights, and the initial marking is replaced by a single token initially in `Idle`.

To show the unfolding of a CPN transition, consider the CPN subnet in Figure 7. To represent it as a PTN, the places must be unfolded, as explained in the previous paragraph and, in addition, the transition `ReceiveCanCommit` must be unfolded to a PTN transition for each of the four bindings listed in Figure 6. Figure 9 outlines the equivalent PTN; arc weights have been omitted, as all are 1; the marking shown is the initial marking.

We have outlined how to unfold two CPN subnets into PTN subnets. For larger color sets, the unfolding yields an explosion in size, since there will be a PTN place for each color of a CPN place and a PTN transition for each binding of a CPN transition. For our CPN model the modeler just needs to change the symbolic constant `W` (defined in Figure 2) to configure the model to handle, say, five workers. With basic Petri nets the modeler would need to add places, transitions, and arcs and

Figure 8. PTN representation of the CPN in Figure 4.

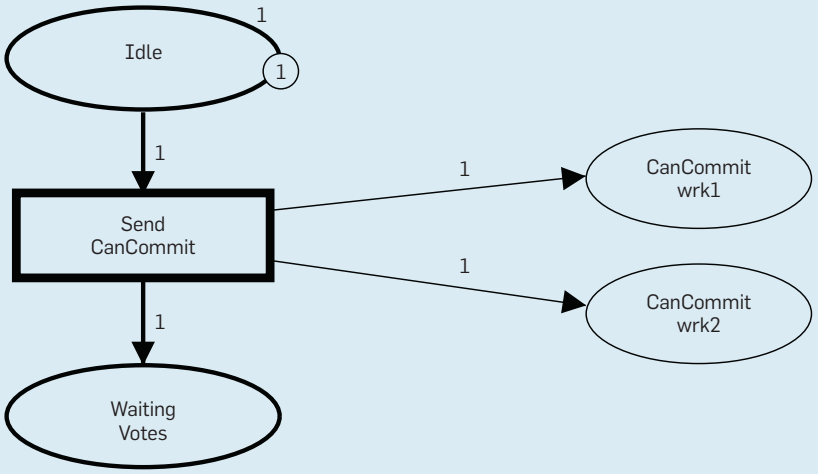
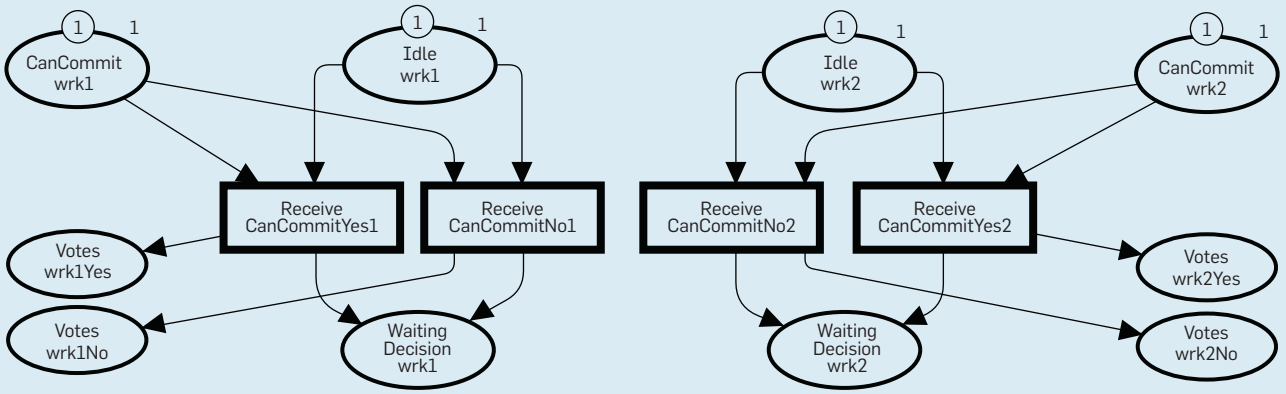


Figure 9. PTN representation of the CPN in Figure 7.



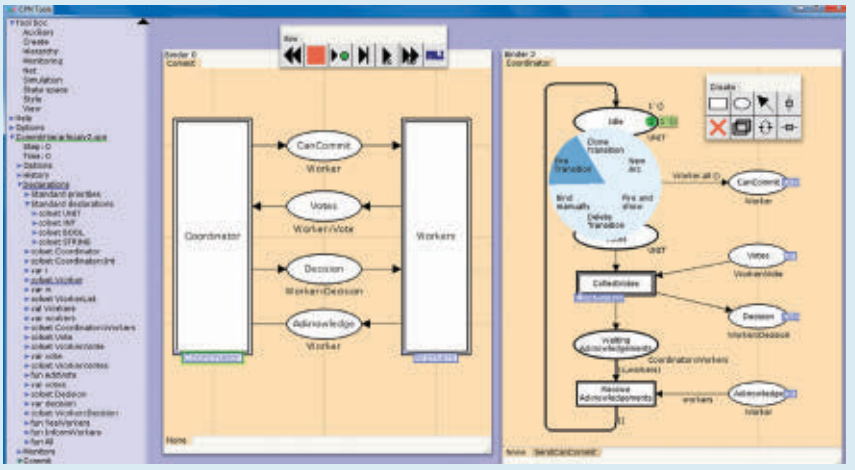
State Spaces and Model Checking

State space exploration is the main technique used for verifying behavioral properties of CPN models. In its basic form, a state space is a directed graph consisting of a node for each reachable marking of the CPN model and edges corresponding to occurrences of enabled bindings. When conducting state space exploration, CPN Tools explores all the enabled transition bindings in each encountered marking. The expressive power of CPNs implies state spaces of CPN models may be infinite. However, many CPN models arising in practice do have a finite state space or can be modified (such as by bounding data types and number of tokens on places) to have a finite state space. With a finite state space, the modeler is able to automatically verify a range of standard behavioral properties of CPN models (such as boundedness, home, and liveness properties). In addition, state spaces of CPN models can be used to perform model checking of behavioral properties specified in temporal logic. The main limitation of state space methods is the inherent state exploration problem, implying the full state space is often too large to be explored with the available computing power. Many advanced techniques exist to combat the state explosion problem, most applicable in the context of CPN models. The state space methods in CPN Tools were developed by Christensen et al.[7,29]

Timed Models and Performance Analysis

CPNs also include a notion of time inspired by van der Aalst,[26] making it possible to model the time taken by different activities. The time concept is based on the introduction of a "global clock" representing the current model time and attaching "time stamps" to tokens, in addition to the token colors. The time stamp of a token specifies the earliest model time at which the token can be removed by the occurrence of a transition, and delay inscriptions on transitions and arcs are used to determine the timestamps on tokens produced by transitions. During model execution, the global model clock is advanced automatically to the earliest next time at which a transition becomes enabled, and the model stays at the current model time until no more transitions are enabled. The time concept provides the foundation for conducting simulation-based performance analysis of CPN models. The basic idea of simulation-based performance analysis is to conduct a number of lengthy simulations during which data on, say, queue length and delays, is collected. Data collection is based on the concept of monitors observing simulations and writing the extracted data into log files for post processing or directly computing key performance figures (such as averages, standard deviations, and confidence intervals). Batch simulations can also be used to explore the parameter space of the model and conduct multiple simulations for each parameter in order to obtain statistically reliable results. Support for performance analysis is based on the work of Wells and Lindstrøm.[27]

Figure 10. The two-phase commit CPN model in CPN Tools.



hence change the net structure. This example shows CPNs—unlike basic Petri nets—provide the means for creating compact parameterizable models.

Tools and Applications

The construction and analysis of CPN models are supported by two generations of software tools: Design/CPN[6] and CPN Tools.[28] They have been instrumental in CPN success, as they enable the practical use in a broad range of domains, including distributed software systems, communication protocols, embedded systems, and process and workflow modeling. Design/CPN and CPN Tools both work directly on the high-level representation of CPN models and do not perform unfolding to the underlying PTN model, as discussed earlier. For a comprehensive list of more than 100 papers describing practical applications and domains see the webpages of the CPN group at Aarhus University, Denmark.[13]

The Design/CPN tool[6] was created by Meta Software, Cambridge, MA, starting in 1988. Kurt Jensen, Robert M. Shapiro, and Peter Huber were the main architects, and the implementation was carried out with an international group of researchers. The first version of Design/CPN supported modeling, syntax checking, and interactive simulation. Introduction of hierarchical CPNs supported by the Design/CPN tool advanced the practical use of Petri nets. The new modeling language and its tool support were general and powerful enough to eliminate the need for making ad hoc extensions, as discussed earlier. A common platform for practical modeling was established and used by most Petri net practitioners, as supported by Jensen's three-volume monograph on CPNs, 1992–1997.[15]

Starting in 2000, a second generation of tool support, called CPN Tools,[28] was designed and implemented at Aarhus University. The main architects were Kurt Jensen, Søren Christensen, and Michael Westergaard. The design was based on empirical studies of the use of Design/CPN, making the new tool much easier and more efficient to use when constructing CPN models. In 2010, CPN Tools had 10,000 licenses in 150 countries.

At the time, development and maintenance of the tool set were transferred to the group of Wil van der Aalst at the Technical University of Eindhoven, the Netherlands.[28] New updates with improved functionality continue to be distributed on a regular basis.

CPN Tools supports editing and construction of CPN models, interactive and automatic simulation, state space-based model checking (see the second sidebar "State Spaces and Model Checking"), and simulation-based performance analysis (see the third sidebar "Timed Models and Performance Analysis"). CPN Tools is based on a much faster simulation engine developed by Bisgaard et al.[20] With this simulation engine, many models run more than 1,000 times faster compared to Design/CPN, allowing complex automatic simulations to be executed within seconds instead of hours.

Figure 10 is a screenshot of CPN Tools with the CPN model of the two-phase commit transaction system. The modeler works directly with the graphical representation of the CPN model through a user interface based on such interaction techniques as "tool palettes" and "marking menus." The rectangular area to the left is an "index" that includes the `Tool box` for the user to manipulate the declarations and modules. The rest of the screen is the "workspace," which, in this case, contains two "binders" (the rectangular windows) and a circular pop-up menu. The binder to the left contains the `Commit` module, and the binder to the right contains the `Coordinator` module. In addition are two tool palettes: one—`Sim`—containing the tools that can be used for simulation of the model, the other—`Create`—containing the tools for creating CPN model elements. A circular marking menu appears on top of a transition to provide the operations that can be performed on a transition.

CPN Tools performs syntax and type checking, and contextual error messages are provided to the user. The syntax check and code generation are incremental and performed in parallel with editing. This means it is possible to execute parts of a CPN model even if the model is not complete and that when parts of a CPN model are modified a syntax check and code generation are performed only on the elements that depend on the parts that were modified. CPN Tools supports two types of simulation: interactive and automatic. In an "interactive" simulation, the user is in complete control and determines the individual steps in the simulation, selecting among the enabled bindings in the current state. CPN Tools shows the effect of executing a selected binding in the graphical representation of the CPN model. In an "automatic simulation," the user specifies the number of steps to be executed and/or sets a number of stop criteria and breakpoints. The simulator then automatically executes the model without user interaction by making random choices among the enabled bindings in the states encountered. Only the resulting state is shown, but information about the executed steps can be collected in various ways. CPN Tools also includes support for domain-specific graphical feedback from simulations developed by Westergaard and Lassen.[30]

The modeler works directly with the graphical representation of the CPN model through a user interface based on such interaction techniques as "tool palettes" and "marking menus."

Conclusion

Research into CPNs is driven by simultaneous focus on theoretical development, design, and implementation of software-tool support, as well as practical application and case studies and their mutual influence on one another. A theoretical foundation is needed to develop semantically sound software tools that are in turn needed for practical applications that reveal limitations of the theoretical foundation.

Creation of the CPN language can be divided into four steps reflecting many similarities with developments in programming languages. The first, from the black tokens of basic Petri nets to colored tokens in PrT nets, corresponds to the step from bit representation to simple data types. The second, from PrT nets to CPNs, corresponds to the introduction of structured data types and type checking. The third, with the introduction of hierarchical CPNs, corresponds to the introduction of structuring concepts like modules, procedures, functions, and subroutines. The fourth, represented by development of Design/CPN and CPN Tools, corresponds to the implementation of

compilers and runtime systems that are essential for practical applications.

In addition to CPNs, several other variants of high-level Petri nets and supporting computer tools have been developed based on the idea of extending Petri nets with data types. One example is Well-Formed Nets (WFNs),[4] as supported by the CosyVerif tool. WFNs put rather strong restrictions on the color sets and associated operations to facilitate space-efficient exploration of state spaces in which the nodes correspond to equivalence classes of states instead of single states. Concurrent Object-Oriented Nets,[2] as implemented in the COOPNBuilder tool, incorporate object-oriented concepts into Petri nets and rely on Algebraic Petri Nets (APNs)[22] for specification of data types and inscriptions. APNs are high-level Petri nets in which algebraic abstract data types are used for giving semantics to the inscriptions. Reference Nets (RNs)[25] are high-level Petri nets targeting agent-oriented object systems in which the tokens themselves may constitute references to other Petri net models. RNs are supported by the Renew tool, using Java as the inscription language. Stochastic Well-formed Nets,[5] as supported by the GreatSPN tool, enable performance evaluation based on Markov chains. Billington[3] developed an international standard for high-level Petri nets that was approved in 2004. The high-level Petri net standard was heavily based on CPNs that conform to the standard.

Here, we have provided an overview and introduction to the CPN language and its tool support and discussed the research development that led to the CPN modeling language as it exists today. The most recent monograph on CPNs was published by Jensen and Kristensen in 2009,[16] providing an in-depth but compact introduction to modeling and validation of concurrent systems through CPNs. It introduced the constructs of the CPN modeling language, presented its analysis methods, and provides a comprehensive road map to the practical use of CPNs. Moreover, it presented selected industrial case studies illustrating the practical use of CPNs for design, specification, simulation, and verification in a variety of application domains. It was aimed at use in university courses[17] and self-study.

Acknowledgments

We are indebted to the many colleagues and researchers who have contributed to the development of CPNs. In addition to those mentioned here, Jawahar Malhotra had the idea to use the Standard ML language, Ole Bach Andersen implemented the graphical interface of Design/CPN, and Søren Christensen implemented the binding inference algorithms. Hartmann Genrich contributed to Design/CPN with knowledge and experience from PrT nets. The graphical user interface of CPN Tools was designed together with Michel Beaudouin-Lafon and Wendy McKay.

References

1. Alur, R. and Dill, D. A Theory of timed automata. *Theoretical Computer Science 126*, 2 (Apr. 1994), 183–235.
2. Biberstein, O., Buchs, D., and Guelfi, N. Object-oriented nets with algebraic specifications: The CO-OPN/2 formalism. In *Concurrent Object-Oriented Programming and Petri Nets - Advances in Petri Nets, Lecture Notes in Computer Science 2001.* Springer-Verlag, Berlin-Heidelberg, 2001, 73–130.
3. Billington, J. *ISO/IEC 15909-1:2004, Software and System Engineering - High-Level Petri Nets - Part 1: Concepts, Definitions and Graphical Notation.* ISO/IEC, Switzerland, Dec. 2004; http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=38225
4. Chiola, G., Dutheillet, C., Franceschinis, G., and Haddad, S. On well-formed coloured nets and their symbolic reachability graph. Chapter 13 in *High-Level Petri Nets*, Springer-Verlag, Berlin-Heidelberg, 1991, 373–396.
5. Chiola, G., Dutheillet, C., Franceschinis, G., and Haddad, S. Stochastic well-formed coloured nets and symmetric modelling applications. *IEEE Transactions on Computers 42*, 11 (Nov. 1993), 1343–1360.
6. Christensen, S., Jørgensen, J.B., and Kristensen, L.M. Design/CPN: A computer tool for coloured Petri nets. In *Proceedings of the Third International Workshop on Tools and Algorithms for Construction and Analysis of Systems* (Enschede, the Netherlands, Apr. 2–4) Lecture Notes on Computer Science 1217. Springer-Verlag, Berlin-Heidelberg, 1997, 209–223.
7. Christensen, S., Kristensen, L.M., and Mailund, T. A sweep-line method for state space exploration. In *Proceedings of the Seventh International Conference on Tools and Algorithms for Construction and Analysis of Systems* (Genova, Italy, Apr. 2–6) Lecture Notes on Computer Science 2031. Springer-Verlag, Berlin-Heidelberg, 2001, 450–464.
8. Genrich, H. and Lautenbach, K. System modelling with high-level Petri nets. *Theoretical Computer Science 13*, 1 (1981), 109–136.
9. Girault, C. and Valk, E., Eds. *Petri Nets for Systems Engineering.* Springer-Verlag, Berlin-Heidelberg, 2003.
10. Harel, D. Statecharts: A visual formalism for complex systems. *Science of Computer Programming 8*, 3 (June 1987), 231–274.
11. Holzmann, G. *The SPIN Model Checker.* Addison-Wesley, 2003.
12. Huber, P., Jensen, K., and Shapiro, R. Hierarchies in coloured Petri nets. In *Advances in Petri Nets 1990, Lecture Notes on Computer Science 483.* Springer-Verlag, Berlin-Heidelberg, 1991, 313–341.
13. Jensen, K. et al. *Industrial Use of Coloured Petri Nets.* CPN group at Aarhus University, Denmark; http://www.cs.au.dk/CPnets/industrialex/
14. Jensen, K. Coloured Petri nets and the invariant method. *Theoretical Computer Science 14*, 3 (1981), 317–336.
15. Jensen, K. *Coloured Petri Nets. Vols. 1–3: Basic Concepts, Analysis Methods, Practical Use. Monographs in Theoretical Computer Science.* Springer-Verlag, Berlin-Heidelberg, 1992–1997.
16. Jensen, K. and Kristensen, L.M. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Springer-Verlag, Berlin-Heidelberg, 2009; http://www.cpnbook.org
17. Kristensen, L.M. and Jensen, K. Teaching modelling and validation of concurrent systems using coloured Petri nets. In *Transactions on Petri Nets and Other Models of Concurrency I, Lecture Notes in Computer Science 5100* (Aug. 2008), 19–34.
18. Marca, D. and McGowan, C. *SADT: Structured Analysis and Design Technique.* McGraw-Hill, New York, 1988.
19. Milner, R., Tofte, M., Harper, R., and MacQueen, D. *The Definition of Standard ML.* MIT Press, Cambridge, MA, 1997.
20. Mortensen, K.H. Efficient data structures and algorithms for a coloured Petri nets simulator. In *Proceedings of the Third Workshop and Tutorial on Practical Use of Colored Petri Nets and CPN Tools* (Aarhus, Denmark, Aug. 29–31). Department of Computer Science, University of Aarhus, Denmark, 2001, 57–74.
21. Petri, C.A. *Kommunikation mit Automaten, Schriften des Instrumentelle Mathematik (IIM) 2, Technical Report.* Institut für Instrumentelle Mathematik, Bonn, Germany, 1962.
22. Reisig, W. Petri nets and algebraic specifications. *Theoretical Computer Science 80*, 1 (Mar. 1991), 1–34.
23. Schiffers, M. and Wedde, H. Analyzing program solutions of coordination problems by CP-nets. In *Proceedings of Seventh Symposium of Mathematical Foundations of Computer Science* (Zakopane, Poland, Sept. 4–8), Lecture Notes in Computer Science 64. Springer-Verlag, Berlin-Heidelberg, 1978, 462–473.
24. Shapiro, R. *Towards a Design Methodology for Information Systems, Technical Report.* In Petri, C.A.: Ansätze zur Organisationstheorie rechnergestützter Informationssysteme, 1978.
25. Valk, R. Object petri nets: Using the nets-within-nets paradigm. In *Advances in Petri Nets, Lecture Notes on Computer Science 3098.* Springer-Verlag, Berlin-Heidelberg, 2004, 819–848.
26. van der Aalst, W. Interval timed coloured Petri nets and their analysis. In *Proceedings of the 14th International Conference on Application and Theory of Petri Nets* (Chicago, June 21–25), Lecture Notes on Computer Science 691. Springer-Verlag, Berlin-Heidelberg, 1993, 453–472.
27. Wells, L.M. Performance analysis using CPN tools. In *Proceedings of the First International Conference on Performance Evaluation Methodologies and Tools* (Pisa, Italy, Oct. 11–13), Vol. 180 of ACM Conference Proceeding Series. ACM Press, New York, 2006, article no. 59.
28. Westergaard, M. et al. CPN tools for editing, simulating, and analysing coloured Petri nets. In *Proceedings of the 24th International Conference on Applications and Theory of Petri Nets* (Eindhoven, the Netherlands, June 23–27) Lecture Notes on Computer Science 2679. Springer-Verlag, Berlin-Heidelberg, 2003, 450–462; see also http://www.cpntools.org
29. Westergaard, M., Evangelista, S., and Kristensen, L.M. ASAP: An extensible platform for state space analysis. In *Proceedings of 30th International Conference on Application and Theory of Petri Nets* (Paris, June 22–26) Lecture Notes on Computer Science 5606. Springer-Verlag, Berlin-Heidelberg, 2009, 303–312.
30. Westergaard, M. and Lassen, K. The BRITNeY suite animation tool. In *Proceedings of the 27th International Conference on Application and Theory of Petri Nets and Other Models of Concurrency* (Turku, Finland, June 26–30) Lecture Notes on Computer Science 4024. Springer-Verlag, Berlin-Heidelberg, 2006, 431–440.

Kurt Jensen (kjensen@cs.au.dk) is a professor in and vice-head of the Department of Computer Science of Aarhus University, Denmark; he was head of the department 1998–2014.

Lars M. Kristensen (lmkr@hib.no) is a professor in the Department of Computing, Mathematics, and Physics at Bergen University College, Norway.

© 2015 ACM 0001-0782/15/06 $15.00

DOI:10.1145/2771951

Leslie Lamport is the recipient of the 2013 ACM A.M. Turing Award.

BY LESLIE LAMPORT

The Computer Science of Concurrency: The Early Years

To Edsger Dijkstra

It is insufficiently considered that men more often require to be reminded than informed.

Samuel Johnson

I don't know if concurrency is a science, but it is a field of computer science. What I call *concurrency* has gone by many names, including parallel computing, concurrent programming, and multiprogramming. I regard distributed computing to be part of the more general topic of concurrency. I also use the name *algorithm* for what were once usually called programs and were generally written in pseudo-code.

This is a personal view of the first dozen years of the history of the field of concurrency—a view from today, based on 40 years of hindsight. It reflects my biased perspective, so despite covering only the very beginning of what was then an esoteric field, it is far from complete. The geneses of my own contributions are described in comments in my publications web page.

The omission that would have seemed most striking to someone reading this history in 1977 is the absence of any discussion of programming languages. In the late 1960s and early 1970s, most papers considered to be about concurrency were about language constructs for concurrent programs. A problem such as mutual exclusion was considered to be solved by introducing a language construct that made its solution trivial. This article is not about concurrent programming; it is about concurrent algorithms and their underlying principles.

The Beginning: Mutual Exclusion

The Problem. While concurrent program execution had been considered for years, the computer science of concurrency began with Edsger Dijkstra's seminal 1965 paper that introduced the mutual exclusion problem.[5] He posed the problem of synchronizing N processes, each with a section of code called its *critical section*, so that the following properties are satisfied:

Mutual Exclusion. No two critical sections are executed concurrently. (Like many problems in concurrency, the goal of mutual exclusion is to eliminate concurrency, allowing us to at least pretend that everything happens sequentially.)

Livelock Freedom. If some process is waiting to execute its critical section, then some process will eventually execute its critical section.

Mutual exclusion is an example of what is now called a *safety* property, and livelock freedom is called a *liveness* property. Intuitively, a safety property asserts that something bad never happens; a liveness property asserts that something good must eventually happen. Safety and liveness were defined formally in 1985.[1]

Dijkstra required a solution to allow any computer to halt outside its critical section and associated synchronizing

code. This is a crucial requirement that rules out simple, uninteresting solutions—for example, ones in which processes take turns entering their critical sections. The 1-buffer case of the producer-consumer synchronization algorithm given below essentially is such a solution for $N = 2$.

Dijkstra also permitted no real-time assumption. The only progress property that could be assumed was *process fairness*, which requires every process that has not halted to eventually take a step. In those days, concurrency was obtained by having multiple processes share a single processor. One process could execute thousands of steps while all other processes did nothing. Process fairness was all one could reasonably assume.

Dijkstra was aware from the beginning of how subtle concurrent algorithms are and how easy it is to get them wrong. He wrote a careful proof of his algorithm. The computational model implicit in his reasoning is that an execution is represented as a sequence of states, where a state consists of an assignment of values to the algorithm's variables plus other necessary information such as the control state of each process (what code it will execute next). I have found this to be the most generally useful model of computation—for example, it underlies a Turing machine. I like to call it the *standard model*.

The need for careful proofs should have become evident a few months later, when the second published mutual exclusion algorithm[9] was shown to be incorrect.[10] However, incorrect concurrent algorithms are still being published and will no doubt continue to be for a long time, despite modern tools for catching errors that require little effort—in particular, model checkers.

The First "Real" Solution. Although of little if any practical use, the bakery algorithm[11] has become a popular example of a mutual exclusion algorithm. It is based on a protocol sometimes used in retail shops: customers take successively numbered tickets from a machine, and the lowest-numbered waiting customer is served next. A literal implementation of this approach would require a ticket-machine process that never halts, violating Dijkstra's requirements. Instead, an entering process computes its own ticket number by reading the numbers of all other synchronizing processes and choosing a number greater than any that it sees.

A problem with this algorithm is that ticket numbers can grow without bound. This should not be a practical problem. If each process chooses a number at most one greater than one that was previously chosen, then numbers should remain well below 2^{128}. However, a ticket number might have to occupy more than one memory word, and it was generally assumed that a process could atomically read or write at most one word.

The proof of correctness of the algorithm revealed that the read or write of an entire number need not be atomic. The bakery algorithm is correct as long as reading a number returns the correct value if the number is not concurrently being written. It doesn't matter what value is returned by a read that overlaps a write. The algorithm is correct even if reading a number while it is changing from 9 to 10 obtains the value 2496.

This amazing property of the bakery algorithm means that it implements mutual exclusion without assuming that processes have mutually exclusive access to their ticket numbers. It was the first algorithm to implement mutual exclusion without assuming any lower-level mutual exclusion. In 1973, this was considered impossible.[4(p88)] Even in 1990, experts still thought it was impossible.[21(question 28)]

One problem remained: How can we maintain a reasonable bound on the values of ticket numbers if a read concurrent with a write could obtain any value? For example, what if reading a number while it changes from 9 to 10 can obtain the value 2^{2496}? A closely related problem is to implement a system clock that provides the current time in nanoseconds if reads and writes of only a single byte are atomic, where a read must return a time that was correct sometime during the read operation. Even trickier is to implement a cyclic clock. I recommend these problems as challenging exercises. Solutions have been published.[12]

A Rigorous Proof of Mutual Exclusion. Previous correctness proofs were based on the standard model, in which an execution is represented as a sequence of states. This model assumes atomic transitions between states, so it does not provide a natural model of the bakery algorithm with its non-atomic reads and writes of numbers.

Before I discuss a more suitable model, consider the following conundrum. A fundamental problem of interprocess synchronization is to ensure that an operation executed by one process precedes an operation executed by another process. For example, mutual exclusion requires that if two processes both execute their critical sections, then one of those operation executions precedes the other. Many modern multiprocessor computers provide a Memory Barrier (MB) instruction for implementing interprocess synchronization. Executing an instruction A then an MB then instruction B in a single process ensures that the execution of A precedes that of B. Here is the puzzle: An MB instruction enforces an ordering of two operations performed by the same process. Why is that useful for implementing interprocess synchronization, which requires ordering operations performed by different processes? The reader should contemplate this puzzle before reading the following description of the *two-arrow* model.

In the two-arrow model, an execution of the algorithm is represented by a set of *operation executions* that are considered to have a finite duration with starting and stopping times. The relations $\longrightarrow$ and $\dashrightarrow$ on this set are defined as follows, for arbitrary operation executions A and B:

$A \longrightarrow B$ is true iff (if and only if) A ends before B begins.
$A \dashrightarrow B$ is true iff A begins before B ends.

It is easy to check that these relations satisfy the following properties, for any operation executions A, B, C, and D:

A1. (a) $A \longrightarrow B \longrightarrow C$ implies $A \longrightarrow C$ ($\longrightarrow$ transitively closed)
(b) $A \not\longrightarrow A$. ($\longrightarrow$ irreflexive)
A2. $A \longrightarrow B$ implies $A \dashrightarrow B$ and $B \not\dashrightarrow A$.
A3. $A \longrightarrow B \dashrightarrow C$ or $A \dashrightarrow B \longrightarrow C$ implies $A \dashrightarrow C$.
A4. $A \longrightarrow B \dashrightarrow C \longrightarrow D$ implies $A \longrightarrow D$.

The model abstracts away the explicit concept of time and assumes only a set of operation executions and relations $\longrightarrow$ and $\dashrightarrow$ on it satisfying A1–A4. (An additional property is needed to reason about liveness, which I ignore here.)

Proving correctness of the bakery algorithm requires some additional assumptions:

- All the operation executions within a single process are totally ordered by $\longrightarrow$.
- For any read R and write W of the same variable, either $R \dashrightarrow W$ or $W \longrightarrow R$ holds.

Each variable in the algorithm is written by only a single process, so all writes to that variable are ordered by $\longrightarrow$. We assume that a read that does not overlap a write obtains the correct value. More precisely, if a read R of a variable satisfies $R \longrightarrow W$ or $W \longrightarrow R$ for every write W of the variable, then R obtains the value written by the latest write W with $W \longrightarrow R$.

With these assumptions, the two-arrow formalism provides the most elegant proof of the bakery algorithm that I know of. I published such a proof of a variant of the algorithm.[14]

The conundrum of the MB command described at the beginning of this section is easily explained in terms of the two-arrow formalism. Suppose we want to ensure that an operation execution A in process p precedes an operation execution D in a different process q—that is, to ensure $A \longrightarrow D$. Interprocess communication by accessing shared registers can reveal only that an operation execution C in q sees the effect of an operation execution B in p, which implies $B \dashrightarrow C$. The only way to deduce a $\longrightarrow$ relation from a $\dashrightarrow$ relation is with A4. It allows us to deduce $A \longrightarrow D$ from $B \dashrightarrow C$ if $A \longrightarrow B$ and $C \longrightarrow D$. The latter two $\longrightarrow$ relations can be ensured by using MB instructions, which enforces $\longrightarrow$ relations between operation executions by the same process.

Producer-Consumer Synchronization

The FIFO Queue. The second fundamental concurrent programming problem to be studied was producer-consumer synchronization. This form of synchronization was used at the hardware level in the earliest computers, but it was first identified as a concurrency problem by Dijkstra in 1965, though not published in this formulation until 1968.[6] Here, I consider an equivalent form of the problem: a bounded FIFO (first-in-first-out) queue. It can be described as an algorithm that reads inputs into an N-element buffer and then outputs them. The algorithm uses three variables:

in The infinite sequence of unread input values.
buf A buffer that can hold up to N values.
out The sequence of values output so far.

A *Producer* process moves values from *in* to *buf*, and a *Consumer* process moves them from *buf* to *out*. In 1965 the algorithm would have been written in pseudo-code. Today, we can write it in the PlusCal algorithm language[15] as algorithm *PC* of Figure 1. The initial value of the variable *in* is the constant *Input*, which is assumed to be an infinite sequence of values; variables *buf* and *out* initially equal the empty sequence. The processes *Producer* and *Consumer* are given the identifiers 0 and 1. In PlusCal, an operation execution consists of execution of the code from one label to the next. Hence, the entire body of each process's **while** loop is executed atomically. The **await** statements assert enabling conditions of the actions. The keywords **fair** specify process fairness.

Figure 2 shows the first four states of an execution of the algorithm represented in the standard model. The letter P or C atop an arrow indicates which process's atomic step is executed to reach the next state.

Algorithm *PC* is a specification; a bounded FIFO queue must implement that specification. A specification is a definition, and it makes no formal sense to ask if a definition is correct. However, we can gain confidence that this algorithm does specify a bounded FIFO queue by proving properties of it. The most important class of properties one proves about an algorithm are invariance properties. A state predicate is an *invariant* iff it is true in every state of every execution. The following invariant of algorithm *PC* suggests that it is a correct specification of an N-element bounded queue:

$$(Len(buf) \leq N) \ \wedge \ (Input = out \circ buf \circ in)$$

where $Len(buf)$ is the length of the sequence *buf* and $\circ$ is sequence concatenation.

The basic method for proving that a predicate *Inv* is an invariant of a concurrent algorithm was introduced by Edward Ashcroft in 1975.[2] We find a suitable predicate *I* (the inductive invariant)

Figure 1. Producer-consumer synchronization.

```
--algorithm PC {
    variables in = Input, out = ⟨⟩, buf = ⟨⟩;
    fair process (Producer = 0) {
      P: while (TRUE) {
           await Len(buf) < N;
           buf := Append(buf, Head(in));
           in  := Tail(in)                    }}
    fair process (Consumer = 1) {
      C: while (TRUE) {
           await Len(buf) > 0;
           out := Append(out, Head(buf));
           buf := Tail(buf)                   }}  }
```

Figure 2. An execution of the FIFO queue.

$$\begin{bmatrix} in = \langle v_1, v_2, \ldots \rangle \\ out = \langle\,\rangle \\ buf = \langle\,\rangle \end{bmatrix} \xrightarrow{P} \begin{bmatrix} in = \langle v_2, v_3, \ldots \rangle \\ out = \langle\,\rangle \\ buf = \langle v_1 \rangle \end{bmatrix} \xrightarrow{C}$$

$$\begin{bmatrix} in = \langle v_2, v_3, \ldots \rangle \\ out = \langle v_1 \rangle \\ buf = \langle\,\rangle \end{bmatrix} \xrightarrow{P} \begin{bmatrix} in = \langle v_3, v_4, \ldots \rangle \\ out = \langle v_1 \rangle \\ buf = \langle v_2 \rangle \end{bmatrix} \xrightarrow{P} \ldots$$

and prove that (i) I is true in every initial state, (ii) I is left true by every step of the algorithm, and (iii) I implies Inv. It is easy to prove that the state predicate above is an invariant of algorithm PC. The appropriate inductive invariant I is the conjunction of this invariant with a predicate asserting that each variable has a "type-correct" value. (PlusCal is an untyped language.)

Another Way of Looking at a FIFO Queue. The FIFO queue specification allows only a single initial state, and executing either process's action can produce only a single next state. Hence the execution of Figure 2 is completely determined by the sequence $P \rightarrow C \rightarrow P \rightarrow P \rightarrow \cdots$ of atomic-action executions. For $N = 3$, all such sequences are described by the graph in Figure 3. The nodes of the graph are called *events*, each event representing an atomic execution of the algorithm step with which the event is labeled. The graph defines an irreflexive partial order $\prec$ on the events, where $e \prec f$ iff $e \neq f$ and there is a path through the graph from event e to event f. For want of a standard term for it, I will call such a partially ordered set of events, in which events are labeled with atomic steps, an *event history*.

This event history describes all sequences of states that represent executions of algorithm PC in the standard model. Such a sequence of states is described by a sequence of infinitely many P and C events—that is, by a total ordering of the events in the event history. A total ordering of these events describes a possible execution of algorithm PC iff it is consistent with the partial order $\prec$. To see this, observe that the downward pointing diagonal arrows imply that the ith P event (which moves the ith input to the buffer) must precede the ith C event (which moves that input from the buffer to the output). The upward pointing diagonal arrows indicate that the ith C event must precede the $(i+3)$rd P event, which is necessary to ensure that there is room for the $(i+3)$rd input value in the buffer, which can hold at most three elements.

We can view the event history of the figure to be the single "real" execution of algorithm PC. The infinitely many different executions in the standard model are artifacts of the model; they are not inherently different. Two events not ordered by the $\prec$ relation—for example, the second C event and the fourth P event—represent operations that can be executed concurrently. However, the standard model requires concurrent executions of the two operations to be modeled as occurring in some order.

Mutual Exclusion vs. Producer-Consumer Synchronization. Producer-consumer synchronization is inherently deterministic. On the other hand, mutual exclusion synchronization is inherently nondeterministic. It has an inherent race condition: two processes can compete to enter the critical section, and either might win.

Resolving a race requires an *arbiter*, a device that decides which of two events happens first.[3] An arbiter can take arbitrarily long to make its decision. (A well-designed arbiter has an infinitesimal probability of taking very long.) Any mutual exclusion algorithm can therefore, in principle, take arbitrarily long to allow some waiting process to enter its critical section. This is not an artifact of any model. It appears to be a law of nature.

Producer-consumer synchronization has no inherent nondeterminism, hence no race condition. It can be implemented without an arbiter, so each operation can be executed in bounded time. It is a fundamentally different class of problem than mutual exclusion.

The FIFO Queue as an N-Process System. The graph in Figure 3 is drawn with two rows, each containing the events corresponding to actions of one of the two processes. Figure 4 is the same graph drawn with three rows. We can consider the three rows to be three separate processes. If we number these rows 0, 1, and 2 and we number the elements in the *Input* sequence starting from 0, then the events corresponding to the reading and outputting of element i of *Input* are in row i mod 3. We can consider each of those rows to be a process, making the FIFO queue a 3-process system for $N = 3$, and an N-process system in general. If we were to implement the variable *buf* with an N-element cyclic buffer, each of these processes would correspond to a separate buffer element.

In the event history model, any totally ordered subset of events can be considered a process. The standard model has no inherent notion of processes. In that model, an execution is just a sequence of states. Processes are an artifact of the way the sequence of states is represented. The set of executions of algorithm PC can also be described by an N-process PlusCal algorithm.

Generalized Producer-Consumer Synchronization. The generalization of producer-consumer synchronization is marked-graph synchronization. Marked graphs were introduced by Holt and Commoner in 1970.[8] A marked graph is a directed graph together with a *marking* that assigns a finite set of indistinguishable tokens to each arc. A node is *fired* in a marking by removing one token from each of its input arcs and adding one token to each of its output arcs (producing a new marking). A *firing sequence* of a marked graph is a sequence of firings that can end only with a marking in which no node may be fired. (By definition of firing, a node can be fired iff it has at least one token on each input arc.)

A marked graph synchronization problem is described by labeling the

Figure 3. An event history for the FIFO queue with = 3.

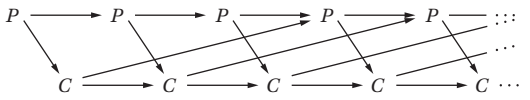
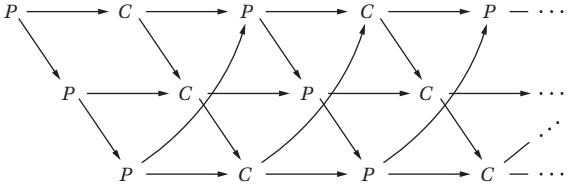
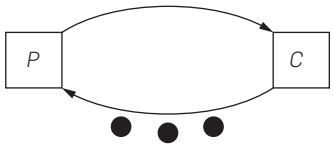


Figure 4. Another view of the FIFO queue for $N = 3$.



nodes of a marked graph with the names of atomic operations. This specifies that a sequence of atomic operation executions is permitted iff it is the sequence of labels of the nodes in a possible firing sequence of the marked graph. For example, the following marked graph describes the FIFO queue for $N = 3$.



A token on the top arc represents a value in the buffer, and a token on the bottom arc represents space for one value in the buffer. Observe that the number of tokens on this marked graph remains constant throughout a firing sequence. The generalization of this observation to arbitrary marked graphs is that the number of tokens on any cycle remains constant.

All executions of a marked graph synchronization algorithm are described by a single event history. Marked graph synchronization can be implemented without an arbiter, so each operation can be executed in a bounded length of time.

Marked graphs can be viewed as a special class of Petri nets.[18] Petri nets are a model of concurrent computation especially well-suited for expressing the need for arbitration. Although simple and elegant, Petri nets are not expressive enough to formally describe most interesting concurrent algorithms. Petri nets have been used successfully to model some aspects of real systems, and they have been generalized to more expressive languages. But to my knowledge, neither Petri nets nor their generalizations have significantly influenced the field of concurrent algorithms.

The Two-Arrow Formalism Revisited. Let $\mathcal{E}$ be an event history with partial order $\prec$. Suppose we partition $\mathcal{E}$ into nonempty disjoint subsets called *operation executions*. We can define two relations $\longrightarrow$ and $\dashrightarrow$ on the set of operation executions as follows, for any operation executions A and B:

$$A \longrightarrow B \text{ iff } \forall\, e \in A, f \in B : e \prec f,$$
$$A \dashrightarrow B \text{ iff } \exists\, e \in A, f \in B : e \prec f.$$

It is straightforward to see that these definitions (and the assumption that $\prec$ is an irreflexive partial order) imply properties A1–A4 on page 72. Thus, we can obtain a two-arrow representation of the execution of an algorithm with non-atomic operations from an event history whose events are the atomic events that make up the operation executions. The event history does not have to be discrete. Its events could be points in a space-time continuum, where $\prec$ is the causality relation introduced by Minkowski.[17]

Distributed Algorithms

Pictures of event histories were first used to describe distributed systems. Figure 5 is an event history that I once used as an illustration.[13] The events come from three processes, with time moving upwards. A diagonal arc joining events from two different processes represents the causality relation requiring that a message must be sent before it is received. For example, the arc from q_4 to r_3 indicates that event q_4 of the second process sent a message that was received by event r_3 of the third process.

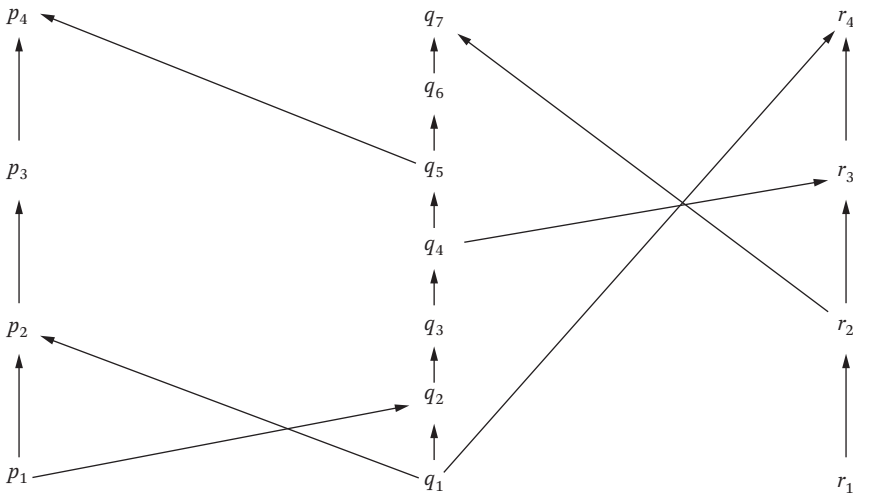
In general, executions of such a distributed system can produce different event histories. For example, in addition to the history of Figure 5, there might be an event history in which the message sent by event q_1 is received before the message sent by event q_4. In such a case, there is true nondeterminism and the system requires arbitration.

Let a *consistent* cut of an event history consist of a set C of events such that for every two events c and d, if event c is in C and $d \prec c$, then d is in C. For example, $\{p_1, q_1, q_2, r_1, r_2\}$ is a consistent cut of the event history of Figure 5. Every consistent cut defines a global state of the system during some execution in the standard model—the state after executing the steps associated with the events in the consistent cut.

An event history like that of Figure 5 allows incompatible consistent cuts—that is two consistent cuts, neither of which is a subset of the other. They describe possible global states that, in the standard model, may not occur in the same execution. This shows that there is no meaningful concept of a unique global state at an instant. For example, there are different consistent cuts containing only events q_1 and q_2 of the second process. They represent different possible global states immediately after the process has executed event q_2. There is no reason to distinguish any of those global states as *the* global state at that instant.

Because the standard model refers to global states, it has been argued that the model should not be used for reasoning about distributed algorithms and systems. While this argument sounds plausible, it is wrong. An invariant of a global system is a meaningful concept because it is a state predicate that is true for all possible global states, and so does not depend on any preferred global states. The problem of implementing a distributed system can often be viewed as that of maintaining a global invariant even though different processes may have incompatible views of what the current state is at any instant.

Figure 5. An event history for a distributed system.



Thinking is useful, and multiple ways of thinking can be even more useful. However, while event histories may be especially useful for helping us understand distributed systems, the best way to reason about these systems is usually in terms of global invariants. The standard model provides the most practical way to reason about invariance.

Afterwards

After distributed systems, the next major step in concurrent algorithms was the study of fault tolerance. The first scientific examination of fault tolerance was Dijkstra's seminal 1974 paper on self-stabilization.[7] However, as sometimes happens with work that is ahead of its time, that paper received little attention and was essentially forgotten for a decade. A survey of fault tolerance published in 1978[20] does not mention a single algorithm, showing that fault tolerance was still the province of computer engineering, not of computer science.

At about the same time that the study of fault-tolerant algorithms began in earnest, the study of models of concurrency blossomed. Arguably, the most influential of this work was Milner's CCS.[16] These models were generally event-based, and avoided the use of state. They did not easily describe algorithms or the usual way of thinking about them based on the standard model. As a result, the study of concurrent algorithms and the study of formal models of concurrency split into two fields. A number of formalisms based on the standard model were introduced for describing and reasoning about concurrent algorithms. Notable among them is temporal logic, introduced by Amir Pnueli in 1977.[19]

The ensuing decades have seen a huge growth of interest in concurrency—particularly in distributed systems. Looking back at the origins of the field, what stands out is the fundamental role played by Edsger Dijkstra, to whom this history is dedicated. C

References

1. Alpern, B., Schneider, F.B. Defining liveness. *Inform. Process. Lett. 21*, 4 (October 1985), 181–185.
2. Ashcroft, E.A. Proving assertions about parallel programs. *J. Comput. Syst. Sci. 10* (February 1975), 110–135.
3. Barros, J.C., Johnson, B.W. Equivalence of the arbiter, the synchronizer, the latch, and the inertial delay. *IEEE Trans. Comput. C-32*, 7 (July 1983), 603–614.
4. Hansen, P.B. *Operating System Principles*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.
5. Dijkstra, E.W. Solution of a problem in concurrent programming control. *Commun. ACM 8*, 9 (September 1965), 569.
6. Dijkstra, E.W. Cooperating sequential processes. In *Programming Languages*. F. Genuys, ed. Academic Press, New York, 1968, 43–112. Originally appeared as EWD123 (1965).
7. Dijkstra, E.W. Self-stabilizing systems in spite of distributed control. *Commun. ACM 17*, 11 (November 1974), 643–644.
8. Holt, A., Commoner, F. Events and conditions. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*. Project MAC, June 1970, 3–52.
9. Hyman, H. Comments on a problem in concurrent programming control. *Commun. ACM 9*, 1 (Jan. 1966).
10. Knuth, D.E. Additional commments on a problem in concurrent program control. *Commun. ACM 9*, 5 (May 1966), 321–322.
11. Lamport, L. A new solution of Dijkstra's concurrent programming problem. *Commun. ACM 17*, 8 (August 1974), 453–455.
12. Lamport, L. Concurrent reading and writing. *Commun. ACM 20*, 11 (November 1977), 806–811.
13. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM 21*, 7 (July 1978), 558–565.
14. Lamport, L. A new approach to proving the correctness of multi-process programs. *ACM Trans. Program. Lang. Syst. 1*, 1 (July 1979), 84–97.
15. Lamport, L. The PlusCal algorithm language. In *Theoretical Aspects of Computing, ICTAC 2009*, Volume 5684 of *Lecture Notes in Computer Science*. M. Leucker and C. Morgan, eds. Springer-Verlag, 2009, 36–60.
16. Milner, R. *A Calculus of Communicating Systems*, Volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Heidelberg, N.Y., 1980.
17. Minkowski, H. Space and time. In *The Principle of Relativity*. Dover, 1952, 73–91.
18. Petri, C.A. Fundamentals of a theory of asynchronous information flow. In *Information Processing 1962, Proceedings of IFIP Congress 62*. C.M. Popplewell, ed. North-Holland, 1962, 386–390.
19. Pnueli, A. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on the Foundations of Computer Science*. IEEE, November 1977, 46–57.
20. Randell, B., Lee, P.A., Treleaven, P.C. Reliability issues in computing system design. *Comput. Surv. 10*, 2 (June 1978), 123–165.
21. Rudolph, B.A. Self-assessment procedure xxi. *Commun. ACM 33*, 5 (May 1990), 563–575.

Copyright held by author.
Publication rights licensed to ACM. $15.00

SIGMOD PODS

San Francisco 2016

ACM SIGMOD/PODS Conference

Hyatt Regency, San Francisco, California, USA
June 26 – July 1, 2016
http://www.sigmod2016.org

Important Dates

SIGMOD 2016 Deadlines

Research paper 1st submission dates

July 9, 2015: Abstract submission
July 16, 2015: Paper submission

Research paper 2nd submission dates

November 12, 2015: Abstract submission
November 19, 2015: Paper submission

PODS 2016 Deadlines

Research paper 1st submission dates

October 2, 2015: Abstract submission
October 9, 2015: Paper submission

Research paper 2nd submission dates

November 27, 2015: Abstract submission
December 4, 2015: Paper submission

Conference Officers

Main SIGMOD 2016 Officers

General Chairs

Fatma Özcan (IBM Research, USA)
Georgia Koutrika (HP Labs, USA)

Program Chair

Sam Madden (MIT, USA)

Main PODS 2016 Officers

General Chair

Tova Milo (Tel Aviv University, Israel)

Program Chair

Wang-Chiew Tan (UC Santa Cruz, USA)

Call for Contributions

ACM SIGMOD is a leading international forum for academia and industry to explore cutting-edge ideas and results, and to exchange techniques, tools, and experiences. It is co-located with PODS focusing on the theoretical aspects of databases.

ACM SIGMOD/PODS 2016 solicits submissions to its different events (research papers, demos, tutorials, workshops etc.). We invite the submission of original contributions relating to all aspects of data management defined broadly, and particularly encourage submissions on topics of emerging interest in the research and development communities. The call for papers and detailed submission information will be available at: http://www.sigmod2016.org/

San Francisco

A unique bridge, cable cars, a sparkling bay, a famous prison, steep streets with beautiful Victorian houses - so many things make San Francisco one of the world's greatest cities to visit. Located in the Bay Area along the Northern California coast, many attractions beyond those in the city can be easily reached: world-famous wine country, including Napa and Sonoma Valley, beautiful beaches, or world's largest trees and wonderful hikes in close National Parks.

The Hyatt Regency

The conference itself will be held in the Hyatt Regency Hotel, located in the financial district in downtown San Francisco. The hotel is very conveniently located close to many activities and landmarks, making this the ideal spot not only for an exciting conference but also for exploring the area.

Follow SIGMOD16 on Twitter
https://twitter.com/acmsigmod2016

Join SIGMOD16 on Facebook
https://www.facebook.com/sigmod2016

DOI:10.1145/2699392

Future Internet testbeds permit experiments not possible in today's public Net or commercial cloud services.

BY MARK BERMAN, PIET DEMEESTER, JAE WOO LEE, KIRAN NAGARAJA, MICHAEL ZINK, DIDIER COLLE, DILIP KUMAR KRISHNAPPA, DIPANKAR RAYCHAUDHURI, HENNING SCHULZRINNE, IVAN SESKAR, AND SACHIN SHARMA

Future Internets Escape the Simulator

STANDARDIZATION OF BASIC underlying protocols such as the Internet Protocol (IP) has enabled rapid growth and widespread adoption of the global Internet. However, standardization carries the attendant risks of reducing variability and slowing the pace of progress. Validation and deployment of potential innovations by researchers in networking, distributed computing, and cloud computing are often hampered by *Internet ossification*, the inertia associated with the accumulated mass of hardware, software, and protocols that constitute the global, public Internet.[24] Researchers simply cannot develop, test, and deploy certain classes of important innovations into the Internet. In the best case, the experimental components and traffic would be ignored; in the worst case, they could disrupt the correct behavior of the Internet. Cloud computing researchers confront a similar dilemma. In order to maintain uniformity and efficiency in their data centers, commercial cloud providers generally do not provide "under the hood" controls that permit modification to the underlying network topology or protocols that comprise the cloud environment.

A clear example of the challenge is apparent to anyone tracking the pace of adoption of IPv6, a relatively modest revamping of IP. Because IPv6 deployment affects components throughout the Internet, years of extensive review, planning, and coordination have been required to ensure a smooth, if slow, transition. For researchers contemplating more fundamental innovations, such as non-IP protocols or new routing approaches, the barriers are correspondingly higher. Accordingly, researchers have been forced to employ compromise measures, such as validating their novel concepts only in simulation, or in modest, isolated laboratory configurations. These environments permit a wide range of experiments, but at the expense of the realism that comes with a large-scale physical deployment.

» key insights

- **By design, the Internet is inhospitable to many classes of experiments that could lead to major advances in networking, resulting in a problem known as *Internet ossification*.**
- **Researchers worldwide are increasingly turning to future Internet and distributed cloud (FIDC) testbeds, such as GENI in the U.S. and FIRE in the E.U., where they can conduct networking, distributed computing, and cloud computing experiments in a distributed laboratory setting.**
- **Although the technology is still in flux, FIDC testbeds are already supporting important research and education initiatives. As these testbeds join together in international federations, their benefits increase combinatorially.**

IMAGE BY OMELCHENKO



Figure 1. Current GENI deployment phase candidate sites.

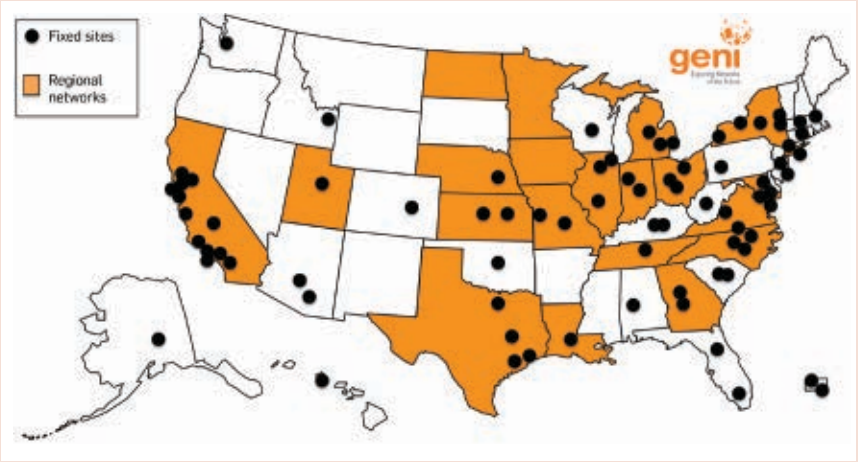
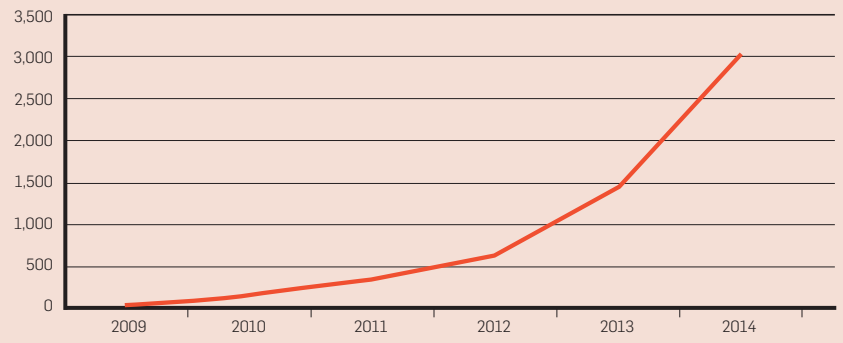


Figure 2. Cumulative unique GENI users.



FIDC Testbeds

Future Internet and distributed cloud testbeds are a promising response to these concerns, providing a virtualized environment where multiple experimental networks may be simultaneously deployed, tested, and validated at significant scale, within a shared platform. These testbeds, beginning with Global Environment for Networking Innovation (GENI) in the U.S.[2] and Future Internet Research & Experimentation (FIRE) in the E.U.,[4] are quickly gaining prevalence and scale. The GENI testbed, for instance, is currently completing its initial deployment phase to 50 sites, with a target of 100–200 sites (see Figure 1).

FIDC testbeds have proven to be successful and versatile in supporting a wide variety of work. For example, as shown in Figure 2, over 3,000 unique users to date have allocated GENI testbed resources for their research and educational work, with the pace of use rapidly increasing. This article is a survey, which introduces key FIDC testbed concepts and presents selected example applications. Interested readers are encouraged to pursue further details found in the referenced documents.

Future Internet testbeds are quickly becoming a global phenomenon supported by a growing international community. Key underlying technologies developed by GENI, FIRE, the University of Utah's Flux group, the OpenFlow and software-defined networking (SDN) communities, the VNode project in Japan, and others are rapidly being combined to form heterogeneous testbeds and interoperable federations. In addition to the U.S. and E.U., national-scale efforts are under way or in planning stages in Japan, Mexico, Canada, China, and South Korea. While these testbeds are built on a variety of underlying technologies, they share certain core capabilities.

Key concepts: Slicing and deep programmability. Establishing a FIDC testbed of meaningful scale requires a significant investment of money and effort. Exclusive use is generally not feasible, so virtualization quickly becomes a practical necessity to support a sizable research community. A key feature of FIDC testbeds is the ability simultaneously to virtualize both computing and networking resources and to assemble them into end-to-end configurations or *slices*. While virtualization of computational resources is reasonably well understood, adding network programmability and virtualization to the mix presents a challenge. However, it is the property of *deep programmability* that creates the key opportunities for innovation in a FIDC testbed. In a deeply programmable environment, the experimenter controls the behavior of computing, storage, routing, and forwarding components deep inside the network, not just at or near the network edge.

As with any virtualized environment, FIDC testbeds present each user with the illusion of exclusive control over shared resources. In a FIDC testbed, the researcher's usual view of his or her resource suite is as a collection of general-purpose computers connected in experimenter-specified topologies by a programmable network, perhaps augmented with special-purpose devices (for example, sensors, high-performance computing resources, and cyberphysical systems). Because these resources are, at their core, real physical computers, networks, and storage devices, rather than simulations, FIDC testbeds can also create powerful opportunities for *end user opt-in*, the ability to run experimental configurations that offer advanced services to real end users.

The selection of virtualization and network programmability approaches represents a trade-off among performance, isolation, and cost. Different testbeds choose different operating points within this space. In fact, some offer multiple virtualization options, even for a single resource type. For example, in GENI a slice's computation resources may include bare metal, Xen, and Vserver hosts, while network programmability options may include modular software routers (for example,

Click), or hardware- or software-based OpenFlow software-defined networking. Similarly, Flare programmable switches in the Japanese VNode network virtualization testbed provide a wide variety of deep programmability options, including efficient native implementations of Click and OpenFlow.

Case Studies

The four efforts discussed here help to illustrate the broad applicability of FIDC testbeds in supporting advanced research and applications. These projects, recently conducted in the GENI testbed and FIRE's iLab.t,[8] apply a wide variety of technology innovations. These range from "clean slate," non-IP protocols to novel cloud computing paradigms and OpenFlow-based software-defined networking. The application domains are similarly diverse, including peer-to-peer message passing, weather forecasting, and video delivery. What these projects share in common is a need to program the underlying network infrastructure in ways not available in the current public Internet.

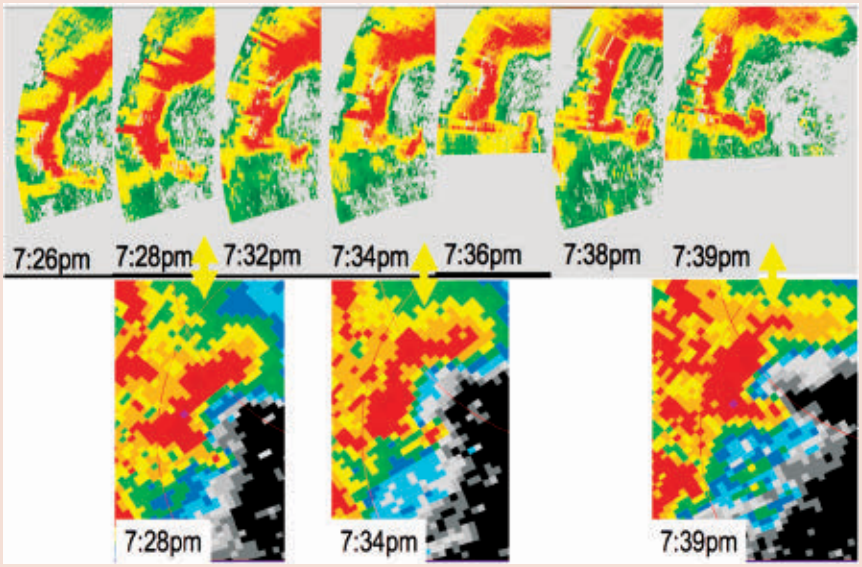
Cloud-based, near-term, localized weather forecasting. CloudCast[10] provides personalized short-term weather forecasts to clients based on their current location using cloud services, generating accurate forecasts tens of minutes in the future for small areas. These short-term "Nowcasts" have profound public safety implications for emergency response to dangerous weather (for example, tornados and severe thunderstorms). Researchers at the U.S. National Science Foundation (NSF) Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere (CASA) are evaluating the benefits and feasibility of providing a large-scale CloudCast service, based on improved weather observations from networks of small, low-cost X-band radars.

An example of the potential benefits of nowcasting is seen in Figure 3, which shows a comparison between data provided by a CASA radar network and data from the current National Weather Service (NWS) NEXRAD system. The upper series of images, from the CASA system, provides much finer temporal and spatial resolution, clearly showing a "hook echo," a rotational event that is a potential indicator of a tornado. CloudCast produces improved forecasts by employing a larger number of relatively short-range radars. These radar networks improve resolution and avoid low-altitude blind spots created over long distances by the curvature of the Earth. Exploiting the data from these sensor networks is computationally intensive, but highly bursty. For example, during a 75-day intensive operation period in the CASA Oklahoma testbed in the spring of 2011, the climatological peak season for severe weather, only 90 hours (or 5% of the period) featured ongoing convective precipitation.

For this reason, the infrastructure-as-a-service (IaaS) model offered by a cloud-computing environment is a very promising approach for a large-scale CloudCast implementation. Instead of acquiring dedicated computing hardware, which would sit idle most of the time, nowcasts could be computed in the cloud, on an as-needed basis. Clearly this economic benefit needs to be balanced with timeliness. Because nowcasts have a very short time horizon, they must be provided to the end users with as little delay as possible. For example, in the case of a 15-minute nowcast, a difference of just a minute or two in nowcast delivery time can be significant. Especially in severe weather situations, providing maximum lead-time can save lives and property.

Experiment design and results. Because timely nowcast generation and delivery relies on a well-engineered combination of network and computational resources, CASA researchers hypothesized that research cloud platforms coupled with control over network assets in an FIDC testbed might outperform commercial clouds. For this investigation, CASA researchers turned to the GENI testbed. They conducted a series of experiments comparing two commercial cloud platforms (Amazon AWS and Rackspace) and two research cloud platforms (GENICloud and ExoGENI), with a goal of improving overall delivery times by reducing

Figure 3. Data from CASA's Oklahoma radar network (top) shows a "hook echo" forming at the center of the image.



Timing results for nowcast algorithm.

Instances	Memory (GB)	Execution time (s)	Total time (s)
Amazon EC2	7.5	74.34	95.08
Rackspace	8	96.53	120.33
GENICloud	8	67.45	78.60
ExoGENI	8	56.83	72.07

data transmission time to and from the cloud computing resources.

Detailed experimental results from four cloud platforms are presented in Krishnappa et al[9] and summarized in the accompanying table. Similar computing resources are reserved on each cloud platform. Execution times are measured by replaying one hour of recorded weather data observed by the CASA radar network in southwestern Oklahoma. The cloud-based nowcast instance receives radar scans, generates 1-minute to 15-minute nowcasts, and stores them on the instance's storage. This continues for one hour, and the average execution time for the generation of each nowcast is measured.

As shown in the table, computation times range from 56.83 seconds on the ExoGENI platform to 96.53 seconds on a Rackspace instance. In comparison to their commercial counterparts, both research cloud instances take less time (67.45 seconds and 56.83 seconds respectively) to compute the nowcasts.

Live process measurement. As a proof of concept for the CloudCast application on cloud services, a live measurement experiment was carried out on each cloud instance to calculate the overall nowcast delivery time from the moment the radar generates the data. These live measurements used a prototype CASA radar located on the University of Massachusetts Amherst campus.[16] Overall results are shown in the last column of the table. These figures provide the most valid information on CloudCast feasibility, because they include the full product delivery chain: raw data transmission time to the instance executing the algorithm, computing time to generate 15-minute nowcast images, and time to transmit the resulting nowcast images to a central Web server for access by end users.

The average overall time for the whole nowcasting process was 95.08 seconds for the EC2 instance, of which 71.98 seconds are consumed in the computation of 15-minute nowcasts on the cloud instance. The remaining 23.10 seconds are used to send data from the radar to the receiving instance, and to transfer the predicted images back to the central server for access by end users. Similarly, the total time taken for executing the whole nowcasting process on Rackspace, GENICloud, and ExoGENI instances is 120.33, 78.60, 72.07 seconds, respectively. Significantly, the dedicated networking resources associated with the research cloud platforms markedly improved overall performance, even without specific tuning for the CloudCast application.

This series of experiments provides convincing evidence of the feasibility of performing short-term weather forecasts in a cloud, with only two minutes to generate and disseminate nowcasts. Additional detailed data on the network and computation components of the overall delivery timeline will provide useful guidance to a potential large-scale CloudCast implementation.

MobilityFirst future Internet architecture. MobilityFirst[20] is a future Internet architecture currently under development as part of the U.S. National Science Foundation (NSF) Future Internet Architecture (FIA) program. The architecture targets a broad set of performance, reliability, and security goals with particular focus on enabling seamless at-scale mobility and establishing trustworthiness as a basic ele-

Figure 4. MobilityFirst protocol stack with GUID service layer providing the new narrow waist.

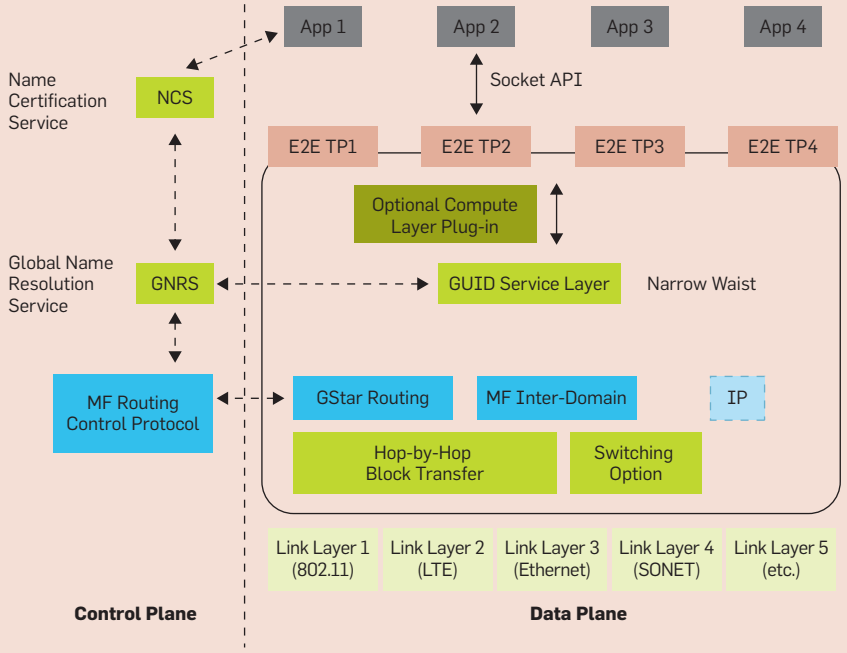
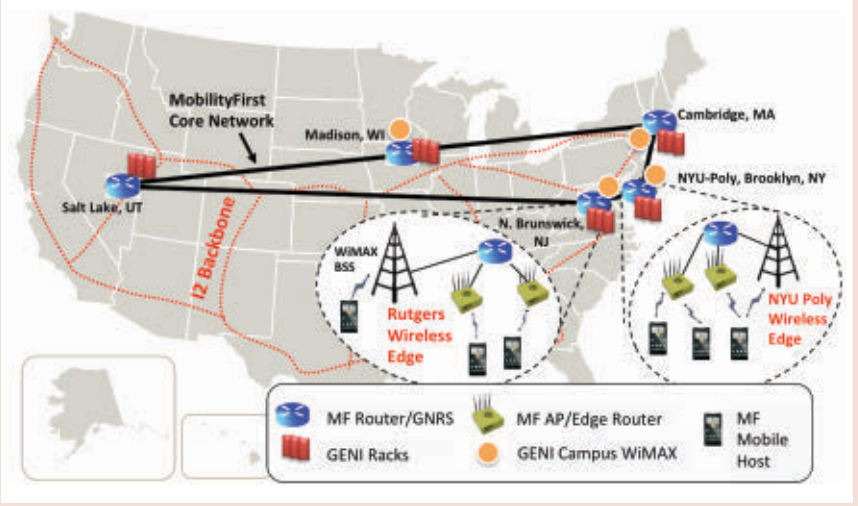


Figure 5. Wide-area deployment of MobilityFirst prototype network on GENI testbed.



ment of network communication in the future Internet. New protocols and in-network services will address wireless access challenges and will provide native support for emerging mobile Internet applications rich in content and context aspects.

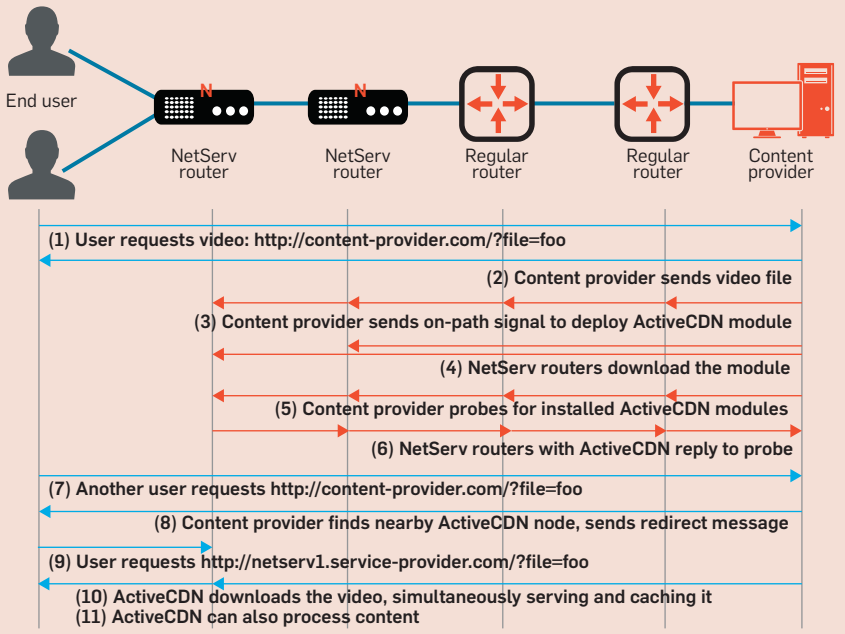
A key design choice made in realizing the preceding goals is to have a logically central naming service at the core of the architecture that is fast, highly scalable, and globally distributed. The naming service includes a name certification service (NCS) that translates human-readable names to unique endpoint identifiers (GUIDs) and a global name resolution service (GNRS) that maps GUIDs to one or more locators (that is, topological or network addresses) of a network-attached object. To enable decentralized trust, the GUID is derived from a public key of an asymmetric key-pair to provide self-certifiable property, that is, a trusted third party is not required to verify identity between two communicating endpoints.[1] Accordingly, the protocol stack (Figure 4) provides name-based networking abstractions primarily by introducing a GUID Service Layer. This new name-based *narrow waist* contrasts with the IP stack in which names and addresses are conflated making seamless mobility a difficult problem. Protocols for scalable inter-domain routing, for reliable data transport, and for scalable multipoint delivery (for example, multicast, anycast, and multihoming), all make use of mappings within GNRS to provide fast dynamic bindings under mobility.[20] *Late-binding*, where only coarse-grain locator information (for example, destination network) is resolved at the packet source, allows packets to be bound at the destination network to the latest address of a mobile endpoint. Data is transported as large blocks in a hop-by-hop manner using a segmented data transport[12] and storage-aware routing[17] protocols that leverage in-network storage to temporarily buffer data under transient problems in wireless access networks such as temporary disconnections under mobility and variable link quality. Finally, a new socket interface provides applications with name-based access to in-network services including direct operations on content (for example, *get* and *put*) and other more abstract objects such as context that can be named using a GUID.[3]

In a deeply programmable environment, the experimenter controls the behavior of computing, storage, routing, and forwarding components deep inside network, not just at or near the network edge.

Evaluation and testbed considerations. To validate these key concepts, the MobilityFirst team chose to build, deploy, and evaluate a prototype MobilityFirst network in a realistic setting. This prototype enables them to characterize the performance and scalability of the GNRS, of routing and transport protocols, and also of end-to-end applications over the MobilityFirst protocol stack. Since protocol behavior and overall performance depend significantly on network properties, a local testbed with limited emulation capabilities alone will fall short. The following capabilities were deemed crucial for the testbed-based evaluations: flexible configurability of network with deep programmability of nodes and network elements; scale of hundreds or more programmable nodes; network with interconnects that are diverse in their latency and bandwidth characteristics; and finally, a broad choice of traditional and emerging network technologies, including wireless (for example, WiFi, 4G WiMAX, and LTE) and SDN (for example, OpenFlow switches).

While the protocol designs were first explored in simulation environments (for example, ns-3 and custom simulator), these were followed up with full-feature prototypes and evaluations in testbed (PlanetLab, GENI) and commercial cloud platforms (Amazon EC2). Prototypes for the key components of the architecture, that is, the naming, routing, and transport services have been implemented. A Click-based software router implements the storage-aware routing and transport services, and closely integrates a GNRS service instance for dynamic name bindings. For the GNRS service, two alternate designs were simultaneously prototyped. One design uses an in-network DHT to store GUID-address mappings (with multiple replicas for each mapping) where service instances are co-located with the routing fabric to minimize access latency.[28] The second design is a flexible overlay implementation, and optimizes service latencies by considering access locality and exerting fine-grained control over replica placement.[27] For end-hosts, the proto-

Figure 6. How ActiveCDN works.



col stack in Figure 4 was implemented for Linux and Android platforms, along with implementations of the new socket API.

Large-scale simulations and limited wide-area end-to-end experiments bear out the benefits of the architecture. Key results include: Both GNRS designs provided 100ms lookup latencies or better when evaluated under current Internet topology assumptions including topologies with up to 26K ASs and 90K links. These latencies could be even smaller (few 10s of ms) in the future if the Internet topology flattens with time (that is, ASs have more direct paths to the core).[28] Limited deployments of GNRS on the GENI testbed across seven rack sites showed 95th percentile latency under 80ms, which mostly reflect the inter-site RTTs; segmented data transport with storage-aware routing significantly improves end-to-end data transfers, particularly under transient wireless access conditions—in some cases an order of magnitude throughput improvements.[12,22] The Click-based router achieves a forwarding rate of approximately 750Mbps on GENI programmable nodes, while the SDN version of the router (OpenFlow/Floodlight with Pronto 3290) achieves close to the 1Gbps line rate.

Wide-area deployment on GENI testbed. Offering extensive heterogeneous resources, geographic diversity, and deep programmability, the GENI testbed was a clear choice for MobilityFirst's goal of at-scale realistic evaluation. Several of the early deployments on GENI were of standalone prototypes and demonstrated the working of key protocols including GNRS and GSTAR in the wide area. Recently, we are maintaining long-running deployment of the more complete prototype network to enable network-level evaluations and to provide an open platform for novel application development. The deployed components include our Click-based router, the in-network GNRS, along with end-hosts running the MobilityFirst stack. It is possible to connect all nodes on a single layer-2 network or establish domains using VLAN programming. The Click routers and GNRS servers are deployed on either bare-metal or VMs, with some as edge-routers having interfaces connected to wireless edge networks with WiFi or WiMAX access. End-hosts are run on GENI testbed nodes or can be user-carried devices (for example, on Android phones and tablets) at GENI campus sites, which can present natural mobile data traffic and mobility events. Experiments so far include validation of in-network DHT version of GNRS, evaluations of reliable content delivery to multihomed mobiles, and a recent demonstration at the 18th GENI Engineering Conference of a novel P2P messaging app that uses name-based networking to address contextual objects like location.

The messaging app—'Drop It'—uses GUIDs to name well-bounded locations such as a conference room or a campus bus stop, while mobile users either 'drop' messages at locations or pick up messages dropped by others. In this pure P2P version, messages remain on originating phones and are retrieved directly when picked up by another phone visiting the location. This is done by maintaining a GNRS mapping from location GUID to the set of phone-GUIDs (or their addresses) that dropped messages at that location, and sending a multicast request to the location GUID, which leverages the hybrid name-address routing in MobilityFirst. This application was demonstrated and evaluated using a multi-site GENI slice (shown in Figure 5) deployment with 10 MobilityFirst software routers (each with GNRS instance), five of which were edge routers providing WiFi and WiMAX client access for Android phones.

Current work and the next phase of the MobilityFirst project are focused on at-scale evaluations and real-world trials of the architecture, including end user opt-in. The plan for GENI-based deployments include footprint of few hundred network elements comprising a core network with several edge networks providing access to clients from participating campus networks or those deployed on third-party cloud platforms (for example, Amazon's EC2).

The GENI deployment will also serve as a crucial evaluation anchor and provide the foundation for three distinct real-world network environment trials in the next phase of the MobilityFirst project. These include a mobile data services trial with a wireless ISP (5Nines) in Madison, WI; a "content production and delivery network" trial involving several public broadcasting stations in Pennsylvania; and a weather emergency notification system based on the CASA radar network described here, with end users in

the Dallas/Fort Worth area. These trials are expected to provide a firm basis for validation of the protocol stack and its utility for advanced mobile, content, context, and cloud applications, while also advancing the technology to the field-deployment stage.

NetServ in-network services architecture. Computing devices connected to the Internet today largely fall into two categories: end systems and routers. The primary goal of a router is to move network packets as quickly as possible. To this end, router vendors typically employ custom operating systems that can take full advantage of the underlying hardware platform. This model limits the ability of a router operator who wishes to customize operation by installing a custom application inside the router—an application developed by a third-party vendor, for instance. So far, the inability to fully customize routers has not hampered the growth of the Internet. Consumer demands have been met by innovative applications running on end systems. The role of routers inside the network has remained the same: moving packets as quickly as possible.

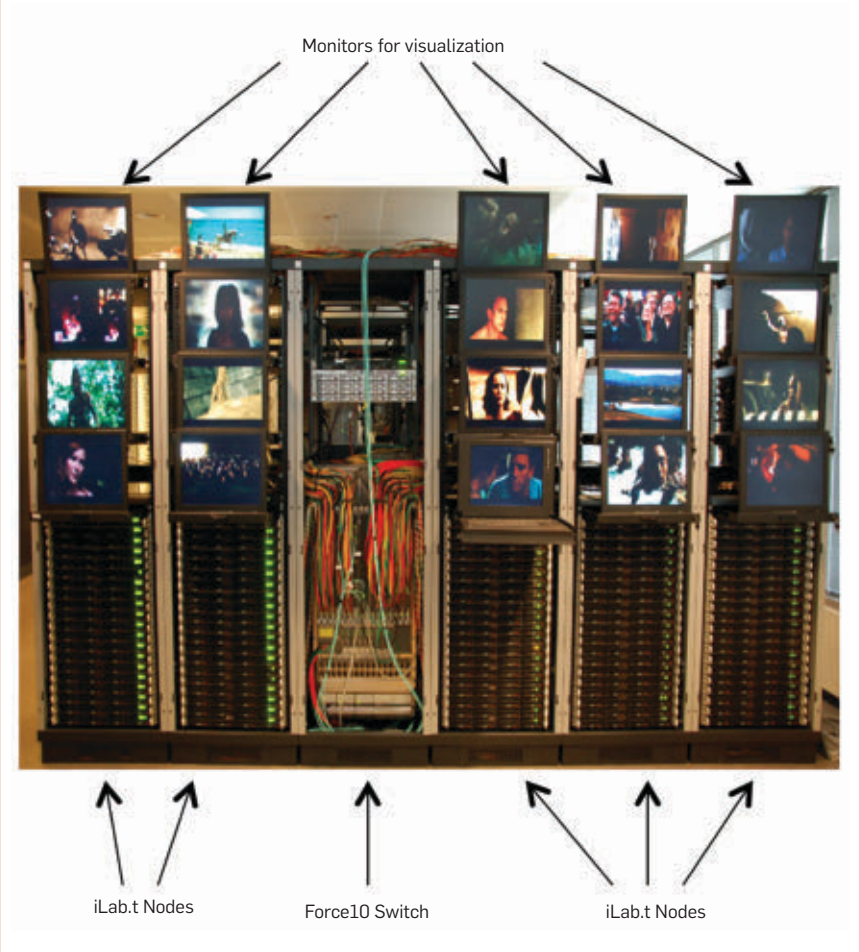
Many signs indicate, however, that the traditional dichotomy between end systems and routers no longer models the complexity of modern networks. With the advent of P2P networking, end systems now frequently assume the role of routers by relaying packets on behalf of other end systems. P2P applications must resort to such overlay networks because routers running fixed software are ill-equipped to form P2P networks, which are usually dynamic, ad hoc, and optimized for specific applications.

Router functionality has not stagnated during this period. Networks grow more complex, ISPs' need and desire for traffic engineering grow more sophisticated, and router vendors respond with new functions like QoS, firewall, VPN, IPsec, NAT, Web cache and rate limiting. Routers have become programmable, but not in the same way that end systems are programmable. Router functions are still limited to a predetermined set supplied by the router vendor itself or the third-party developers approved by the router vendor. In fact, the closed nature of mainstream routers brought the proliferation of middleboxes—network devices that are daisy-chained with routers to provide added functionality—adding to ISPs' network management problems.

Architecture goals. The NetServ team envisions a future Internet where all nodes support a common runtime environment, which eliminates the distinction between end systems and routers for the purpose of running network services.[5,11,14] Network services run on any node, from backbone routers to set-top boxes, and the services are dynamically installed, removed, and migrated to optimal locations on the Internet. As a first step toward that vision, NetServ provides an architectural framework for dynamically deploying in-network services on edge routers. Modules implementing various network functions can be installed at runtime on any NetServ-enabled router on the Internet.

NetServ has adopted five goals in designing a viable in-network service framework. First, NetServ must enable new classes of economically compelling applications that cannot be achieved with existing middleboxes and end systems. One such application is ActiveCDN, which has been demonstrated running on a set of NetServ nodes deployed in the GENI testbed. ActiveCDN is a use case that shows how NetServ can facilitate an economic alliance between ISPs and content providers. Second, NetServ provides a unified runtime environment. A NetServ module can act as an end system application engaged in client-server networking, as a router add-on performing deep packet inspection, or as both at the same time. Third, NetServ provides a wide-area deployment mechanism, using a standardized on-path signaling protocol.[6] A signaling packet is sent to a network destination and the packet is routed as

Figure 7. Setup for failure recovery experiment on ILab.t facility at iMinds.



usual by the regular IP routers. When the signaling packet passes through NetServ-enabled routers, however, the packet will trigger module installations on the routers. Fourth, NetServ provides a multi-user execution environment by running modules inside Java Virtual Machines (JVMs) allocated per user. Fifth, in order to address the performance overhead of running modules in JVM, NetServ has proposed a scalability solution using OpenFlow.

NetServ on GENI testbed. FIDC testbeds provide an ideal platform to deploy and test NetServ. NetServ requires deep programmability to deploy in-network packet processing modules. While emulation-based environments like Mininet and Emulab offer localized deep programmability, a sizable FIDC testbed creates additional experiment options, including measurements of signaling latencies arising from real geographic distances and network topologies. GENI has been an integral part of the NetServ project, in continual use for developing, testing and demonstrating NetServ functionality. The NetServ team has demonstrated two NetServ applications, ActiveCDN and Overload Control,[11] running on GENI at the 9th GENI Engineering Conference.[a]

Researchers have also investigated an autonomic management solution based on NetServ,[5,14] including applying NetServ's dynamic in-network service deployment capability to the problem of counteracting a DoS attack. In one example, an experiment performed on GENI showed a flow-based intrusion detection system was able to reconfigure itself and deploy protection modules quickly using NetServ and OpenFlow, effectively counteracting a DoS attack on a VoIP application server.

ActiveCDN. Figure 6 illustrates a NetServ module deployment scenario using ActiveCDN, a NetServ application that implements CDN functionality on edge routers. When a content provider's server receives a large number of requests for certain content from a particular network area, the server can deploy an ActiveCDN module in the NetServ-enabled routers near the area. The ActiveCDN module then handles subsequent requests for the same content from the network vicinity.

Unlike traditional CDNs, the content provider controls where an ActiveCDN module is deployed. The module can be redeployed to different locations depending on the current traffic. Moreover, the module can perform custom content processing, like inserting region-specific advertisements into video streams.

Many signs indicate the traditional dichotomy between end systems and routers no longer models the complexity of modern networks.

Resilience in OpenFlow networks. Software-defined networking (SDN) decouples the control plane from the data/forwarding plane (switches or routers) of a network and embeds it into one or more external servers called controllers. OpenFlow is the reigning SDN technology, and research teams worldwide investigate many of the research challenges behind it.

The Split Architecture Carrier-Grade Networks (SPARC) project, funded through the European Seventh Framework Programme (FP7), investigated the potential applicability of OpenFlow in carrier-grade telecom networks. Carrier-grade telecom networks support hundreds of thousands of customers, assume failure recovery within 50ms (RFC 5654), and have to deliver high Quality of Experience (QoE) to their customers.

Two well-known recovery techniques, restoration and path protection,[26] are implemented in Openflow.[21] In case of restoration, the controller establishes an alternative path after a failure is detected in the network. In case of path protection, the controller establishes a disjoint alternate path together with the working path. After a failure is detected in the working path, the ingress switch redirects affected traffic to the alternate path using the fast-failover group table implementation of OpenFlow.[19]

Experiment design. The SPARC project has chosen a mixed-fidelity experimentation approach (accurately representing key parts of a system, while simplifying less important parts), using the FIRE testbed to investigate strategies and implementations for failure recovery in challenging network conditions. This approach, combining emulated topologies with realistic output devices, has two prime benefits. The first is the ability to evaluate the actual software stack, including Ericsson OpenFlow software,[18] using real video traffic and displays, simply by

a The 14-minute demo video is at http://vimeo.com/16474575.

observing the quality of video on the displays. Performing such experiments in simulation is very difficult, requiring simulation not only of network functions, but also the software stack of the video codec, the rendering pipeline, and post-processing functions of the video client. The second benefit is the opportunity to create a wide variety of pan-European topologies, controlled network traffic, and failure scenarios to evaluate thoroughly the performance requirements and implementation of switches and controller software. Experiments detailed in Sharma et al.[21] indicate a protection strategy is required to meet recovery time targets and validate the specific failure recovery approach. Although most critical components were tested in a realistic manner, further steps toward deployment will require further testing, for example, on real hardware switches.

Failure recovery experiments are conducted on the iLab.t virtual wall facility at iMinds[8] (see Figure 7). Currently, iLab.t has three virtual walls, each consisting of 100 nodes (multiprocessor, multi-core servers with up to eight 1Gb/s interfaces per server) interconnected by non-blocking Ethernet switches. iLab.t is based on Emulab software developed at the University of Utah, and researchers can build experiments by drawing a topology in a graphical user interface (GUI) or by defining it in scripts. Furthermore, a display wall (20 monitors) is present for easy visualization. iLab.t is part of the Fed4FIRE federation.[25]

Four kinds of failure recovery experiments are performed on the iLab.t facility: validation experiments in one of the pan-European topologies (shown in Figure 8) considering link failures; experiments considering link and node failures on different topologies; scalability experiments in terms of traffic load; and experiments for measuring subjective quality of video.

In the validation experiment, each client sends packets to all other clients at a constant 6ms interval. The experimenters intentionally break the London-Amsterdam link at time zero and observe the failure recovery time for both restoration and protection. Figure 9 shows the traffic destined to the London client, which is going through the Paris-London link, just before and

Figure 8. Emulated failure recovery experiment topology.

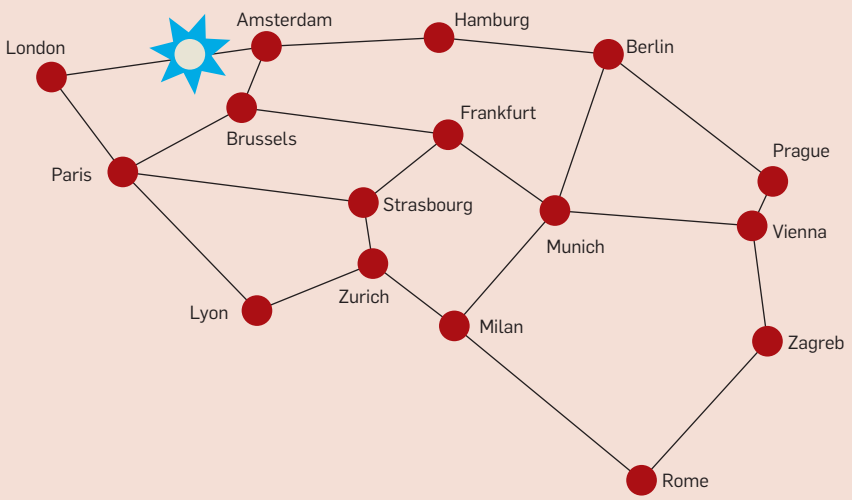


Figure 9. Failure recovery experiment results

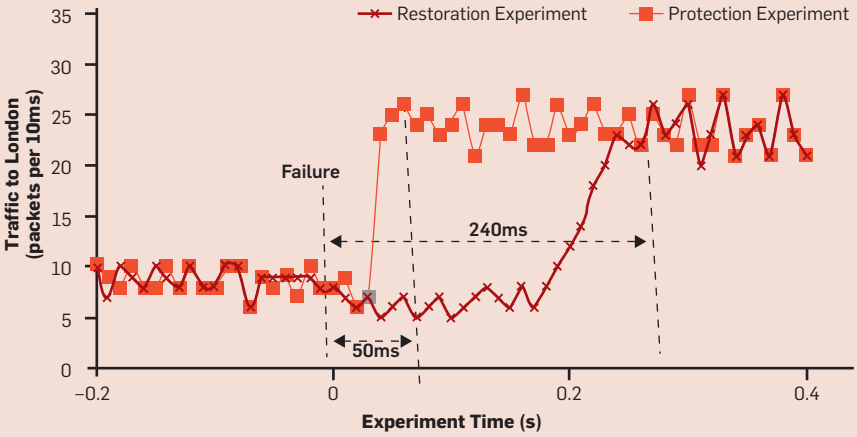


Figure 10. Ongoing FIDC testbed activity.



after the failure. After the failure (at time zero), this is the only link connecting London to the network. Therefore, after the failure all traffic to and from London traverses this link. Total restoration time is around 240ms and the total protection time around 50ms. This result shows that meeting the carrier-grade requirement of 50ms will be very difficult using a centralized controller. Protection is the only choice to meet this stringent requirement.

Failure recovery results. To evaluate the impact of the topology on the recovery time, different pan-European topologies with varying numbers of nodes and node degree are evaluated. The restoration time increases with the number of nodes because the path calculation time grows as $O(n^2)$, where n is the number of nodes. When node degree is reduced, restoration time increases because more hops are used for the restoration path and the controller needs to configure more switches. Protection does not require controller intervention and is therefore far less dependent on the network topology.

The scalability experiment shows an approximately linear relationship between the restoration time and the number of affected flows. In the protection case, the group table implementation effectively mitigates the dependency on the number of affected flows. In all experiments, protection achieves the carrier-grade recovery requirement.

In the video experiment, a video is streamed over the Real-Time Transport Protocol (RTP) and the effect of failure recovery on the video quality is assessed on the display wall. The video is sent using different maximum transmission unit (MTU) sizes. The restoration experiments show clear errors in the video, however, without a clear influence of the MTU. In contrast, protection is so fast that no observable artifacts are noticed.

FIDC Testbeds in Education

An increasing number of universities are turning to FIDC testbeds to support classroom exercises and research projects in their computer science curricula. Perhaps the most straightforward application is to adopt a FIDC testbed as a virtual laboratory for classroom networking experiments. Laboratory-based instruction is already a popular approach for introducing basic networking concepts. A leading example is Liebeherr and El Zarki's classroom networking laboratory design, with companion exercises.[13] A networking student following this approach is quickly exposed to important concepts and tools, such as datagrams, address resolution, configuring a basic IP network and debugging with tools like ping, tcpdump, and wireshark.

Instructors employing a laboratory approach find the transition to classroom experimentation in a FIDC testbed relatively straightforward. Using a virtual laboratory brings many of the same benefits as a physical networking laboratory, with the obvious exception of familiarization with physical networking components. Instructors are attracted to a virtual laboratory because it has greater elasticity for varying class sizes and eliminates the equipment cost (Liebeherr and El Zarki estimate U.S. $1,000 per student) and administrative burden to configure and maintain a classroom laboratory. In addition, FIDC testbeds by their very nature are well suited to manipulation of network settings and configurations that are generally not available to users of standard cloud computing services. Thus, instructors can easily offer basic exercises such as configuring IP addressing and routing for a simple network configuration.

FIDC testbeds have also gained popularity in non-networking courses One example is the FORGE initiative,[15] which is implementing an educational layer over the FIRE testbed facilities, in support of a broad suite of CS laboratory exercises. Another is Williams College Professor Jeannie Albrecht's undergraduate distributed systems class, where students use the GENI testbed to build distributed applications like a Web server, an online bookstore, and a P2P file-sharing system. In addition to simplifying lab setup for the course staff, students see a benefit of using FIDC testbeds for these assignments (compared to local resources), because alternative network topologies with varying conditions can be easily created. Students are then better able to observe the impact of the network on distributed applications.

Educational applications range from relatively basic exercises (for example, configuring IP networking in a small network or implementing a rudimentary Web server) to advanced experiments with novel protocols and individual research projects. Increased exposure to FIDC testbeds in university curricula is raising the comfort level of the next generation of researchers and practitioners who may benefit from these capabilities.

Future Trends and Challenges

FIDC testbeds are already creating additional opportunities for experimental investigations that are not possible in Internet-based and commercial cloud environments. Although the contributions of FIDC testbeds are clear, there remain important unanswered questions in designing and managing these community resources. Many of these questions are fundamental design tensions, such as slicability vs. fidelity, that have been anticipated since the first discussions of GENI's design.[7] Such questions do not have correct answers, but rather describe a spectrum of design options actively being sampled by various testbed developers. As a result, FIDC testbeds such as FIRE and GENI are generally heterogeneous federations of collaborating resource providers, providing experimenters with a variety of options. For example, one experiment may choose to emphasize reproducibility, while another seeks exposure to "in the wild" network traffic.

Other challenges do not arise from deep philosophical conundrums, but simply represent the relatively immature state of these testbeds and their associated tools. While most FIDC researchers are successful in conducting experiments they could not run elsewhere, the user experience can be challenging. There is much good work still to be done to provide both novice and experiment researchers with appropriate tools to design and manage their experiments, particularly those that are long-lived or conducted at large scale. Similarly, tools and processes supporting FIDC infrastructure owners' monitoring and maintenance are often relatively basic.

The trend toward heterogeneous federation of research infrastructure is likely to continue, as testbed developers and their researchers seek greater

scale, flexibility, and variety by joining together in global federations. Figure 10 highlights (in blue) countries and regions worldwide with active FIDC testbed communities. While this trend is exciting, it also creates new challenges in federated policy management and enforcement.

Perhaps the leading example of an emerging global FIDC federation is evident from a collaboration that began in the summer of 2013. Participating testbeds are jointly developing a federation compatibility interface for "clearinghouse" functions, such as user credentialing and project membership, and adopting the GENI "aggregate manager" API for resource discovery and provisioning functions. Federation testing and demonstration began in late 2013, with initial participants including GENI in the U.S., FIRE in the E.U., FIBRE in Brazil and Europe, VNode in Japan, and NICTA in Australia. A monitoring capability, developed at iMinds, continually assesses and reports the status and availability of federation resources, for use by participating experimenters.

FIDC testbeds are gaining global traction, with support from a growing international community. For a variety of practical reasons, testbed developers are increasingly turning to federated designs as a strategy to achieve scale while controlling cost and administrative effort. Federated testbeds rely on a web of trust relationships, uniting the key testbed stakeholders: infrastructure providers, testbed developers, and research/educational users. Several national-scale FIDC testbeds are employing a federation approach to their development and deployment. In addition, a number of these national testbeds are entering into collaborative efforts to create a worldwide, federated infrastructure that facilitates transcontinental research.

Acknowledgments

GENI and CASA are funded by the NSF under cooperative agreements CNS-0737890 and EEC-0313747, respectively. Authors from Columbia, University of Massachusetts, and Rutgers acknowledge support under NSF grants CNS-0831912, CNS-1238485, and CNS-1040735 and CNS-1345295, respectively. Authors Zink and Krishnappa acknowledge support from the GENI project office. Any opinions, findings, conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect the views of the NSF.

Part of this work was carried out with the support of the Fed4FIRE and SPARC projects, funded by the European Commission through the 7th ICT Framework Programme. It does not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

References

1. Anderson, D.G. et al. Accountable Internet Protocol (AIP). *ACM SIGCOMM Computer Communication Review 38*, 4 (2008), 339–350.
2. Berman, M. et al. GENI: A federated testbed for innovative network experiments. *Computer Networks 61* (2014), 5–23.
3. Bronzino, F., Nagaraja, K., Seskar, I. and Raychaudhuri, D. Network service abstractions for a mobility-centric future internet architecture. In *Proceedings of the International Workshop on Mobility in the Evolving Internet Architecture*, (2013), 5–10.
4. Fdida, S. et al. FIRE Roadmap Report 1 – Part II. *Future Internet Research and Experimentation* (FIRE), 2011.
5. Femminella, M., Francescangeli, R., Reali, G., Lee, J.W. and Schulzrinne, H. An enabling platform for autonomic management of the future internet. *IEEE Network 25*, 6 (2011), 24–32.
6. Fu, X., Schulzrinne, H., Bader, A., Hogrefe, D., Kappler, C. and Karagiannis, G. et al. NSIS: A new extensible IP signaling protocol suite. *IEEE Communications 43*, 10 (2005), 133–141.
7. GENI design principles. *Computer 39*, 9 (Sept. 2009), 102–105.
8. iLab.t at iMinds; *http://www.iminds.be/en/develop-test/ilab-t*.
9. Krishnappa, D.K., Lyons, E., Irwin, D. and Zink, M. Network capabilities of cloud services for a real time scientific application. In *Proceedings of the 2012 IEEE 37th Conference on Local Computer Networks*, (2012), 487–495.
10. Krishnappa, D., Irwin, D., Lyons, E. and Zink, M. CloudCast: Cloud computing for short-term weather forecasts. *Computing in Science & Engineering 15*, 4 (2013), 30–37.
11. Lee, J.W. Towards a Common System Architecture for Dynamically Deploying Network Services in Routers and End Hosts.
12. Li, M., Agrawal, D., Ganesan, D., Venkataramani, A. and Agrawal, H. Block-switched networks: A new paradigm for wireless transport. In *Proceedings of Symposium on Networked Systems Design and Implementation 9*, (2009), 423–436.
13. Liebeherr, J. and El Zarki, M. *Mastering Networks: An Internet Lab Manual.* Addison-Wesley, Reading, PA, 2003.
14. Maccherani, E., Femminella, M., Lee, J., Francescangeli, R., Janak, J., Reali, G., et al. Extending the NetServ autonomic management capabilities using OpenFlow. *Network Operations and Management Symposium* (2012), 582–585.
15. Marquez-Barja, J.M., Jourjon, G., Mikroyannidis, A., Tranoris, C., Domingue, J. and DaSilva, L.A. FORGE: Enhancing elearning and research in ICT through remote experimentation. In *Proceedings of the IEEE Global Engineering Education Conference*. Istanbul, Turkey, 2014.
16. McLaughlin, D., Pepyne, D., Chandrasekar, V., Brenda Philips, J. K., Zink, M., Droegemeier, K., et al. Short wavelength technology and the potential for distributed networks of small radar system. *Bulletin of the American Meteorological Society 90*, 12 (2009), 1797-1817.
17. Nelson, S., Bhanage, G., & Raychaudhuri, D. (2011). GSTAR: Generalized storage-aware routing for MobilityFirst in the future mobile Internet. *Proceedings of the 6th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)* ACM, New York, 19–24.
18. OpenFlow Ericsson implementation; https://github.com/TrafficLab/of11softswitch .
19. OpenFlow Specification 1.1; http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf.
20. Raychaudhuri, D., Nagaraja, K. and Venkataramani, A. MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet. *ACM Mobile Computing and Communications Review 16*, 3 (2012), 2–13.
21. Sharma, S., Staessens, D., Colle, D., Pickavet, M. and Demeester, P. OpenFlow: Meeting carrier-grade recovery requirements. *Computer Communications 36*, 6 (2013), 656–665.
22. Somani, N., Chanda, A., Nelson, S.C. and Raychaudhuri, D. Storage aware routing protocol for robust and efficient services in the future mobile Internet. In *Proceedings of the IEEE International Communications Conference, FutureNet V Workshop* (2012).
23. Staelens, N., Deschrijver, D., Vladislavleva, E., Vermeulen, B., Dhaene, T. and Demeester, P. Constructing a no-reference H.264/AVC bitstream-based video quality metric using genetic programming-based symbolic regression. *IEEE Transactions on Circuits and Systems for Video Technology 23*, 8 (2013), 1322–1333.
24. Turner, J. and Taylor, D. Diversifying the Internet. In *Proceedings of the Global Telecommunications Conference* (2005). IEEE.
25. Vandenberghe, W., Vermeulen, B., Demeester, P., Willner, A., Papavassiliou, S. and Gavras, A. et al. Architecture for the heterogeneous federation of future Internet experimentation facilities. In *Proceedings of the Future Network and Mobile Summit 2013 Conference*.
26. Vasseur, J.P., Pickavet, M. and Demeester, P. *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS.* Morgan Kaufmann, 2004.
27. Venkataramani, A., Sharma, A., Tie, X., Uppal, H., Westbrook, D. and Kurose, J. et al. Design requirements of a global name service for a mobility-centric, trustworthy internetwork. *Communication Systems and Networks* (2013) 1–9.
28. Vu, T., Baid, A., Zhang, Y., Nguyen, T.D., Fukuyama, J., Martin, R.P. et al. DMap: A shared hosting scheme for dynamic identifier to locator mappings in the global internet. *Distributed Computing Systems* (2012), 698–707.

Mark Berman (mberman@acm.org) is Vice President for Technology Development and GENI Project Director at Raytheon BBN Technologies, Cambridge, MA.

Piet Demeester (piet.demeester@intec.ugent.be) is a professor at Ghent University–iMinds, Gent, Belgium.

Jae Woo Lee (jae@cs.columbia.edu) is a lecturer in computer science at Columbia University, New York, NY.

Kiran Nagaraja (kiran.nagaraja@ericsson.com) is a senior researcher at Ericsson, Inc. This work was done while he was a senior research associate at WINLAB, Rutgers University, New Brunswick, NJ.

Michael Zink (zink@ecs.umass.edu) is an assistant professor at the University of Massachusetts, Amherst, MA.

Didier Colle (Didier.Colle@UGent.be) is a professor at Ghent University–iMinds, Gent, Belgium.

Dilip Kumar Krishnappa (dilipkuk@akamai.com) is a senior performance engineer, Akami Technologies. This work was done while he was a student at the University of Massachusetts, Amherst, MA.

Dipankar Raychaudhuri (ray@winlab.rutgers.edu) is Distinguished Professr of Electrical and Computer Engineering and Director of WINLAB at Rutgers University, New Brunswick, NJ.

Henning Schulzrinne (hgs@cs.columbia.edu) is Julian Clarence Levi Professor of Computer Science at Columbia University, New York, NY.

Ivan Seskar (seskar@winlab.rutgers.edu) is associate director of WINLAB at Rutgers University, New Brunswick, NJ.

Sachin Sharma (sachin.sharma@intec.ugent.be) is a Ph.D. student at Ghent University–iMinds, Gent, Belgium.

Copyright held by authors.
Publication rights licensed to ACM. $15.00

[RE]VOLUTIONARY



CALL FOR SUBMISSIONS

Submit your work & be a presenter at SIGGRAPH Asia!

SIGGRAPH Asia 2015 invites you to submit your works and showcase your outstanding creative ideas and innovations at the 8th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia, taking place from 2 – 5 November 2015 in Kobe, Japan.

Log-on to **sa2015.siggraph.org/en/submitters.html** to submit your works.

Conference Programs' Submission Deadlines*:

27 May 2015	**Art Gallery**
2 June 2015	**Computer Animation Festival**
10 May 2015	**Courses**
9 June 2015	**Emerging Technologies**
9 June 2015	**Posters**
9 June 2015	**Technical Briefs**
2 June 2015	**Technical Papers**
12 June 2015	**Symposium on Education** NEW!
12 June 2015	**Symposium on Mobile Graphics & Interactive Applications**
12 June 2015	**Symposium on Visualization in High Performance Computing** NEW!
28 April 2015	**Workshops (Proposals Submission Deadline)**
23 June 2015	**Workshops (Papers Submission Deadline)**

* The submission time for all dates is 23:59 UTC/GMT

CALL FOR EXHIBITORS & SPONSORS

Be a part of the SIGGRAPH Asia Exhibition – Asia's Digital Media Marketplace

Meet over 7,000 technical and creative industry experts and individuals from over 50 countries and regions face-to-face to explore business opportunities, partnerships, and to strengthen existing relationships – all in person at SIGGRAPH Asia 2015. Book your stand now by 30 June 2015 to enjoy early bird rates!

Contact **Wyatt Lee, CEM** at **+65 6500 6725** or **wyatt.lee@siggraph.org** for more information on the exhibit space options and fees, as well as sponsorship packages.

CONFERENCE 2 - 5 November 2015
EXHIBITION 3 - 5 November 2015
KOBE CONVENTION CENTER, KOBE, JAPAN
SA2015.SIGGRAPH.ORG

Sponsored by
ACM SIGGRAPH

research highlights

P. 92
Technical Perspective
Virtual Reality in Your Living Room
By Patrick Baudisch

P. 93
IllumiRoom: Immersive Experiences Beyond the TV Screen
By Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson

DOI:10.1145/2754393

To view the accompanying paper, visit doi.acm.org/10.1145/2754391 rh

Technical Perspective
Virtual Reality in Your Living Room

By Patrick Baudisch

VIRTUAL REALITY IS taking off; indeed, it is on the verge of mass availability. But which of the many form factors and underlying principles will it be?

The late 1960s were a defining moment in the long evolution of virtual reality. At that time, Ivan Sutherland explored his dream of an immersive world. By creating the world's first head-worn display driven by 3D graphics, he opened a window into a computer-generated world. He combined his 'Sword of Damocles' display system with a method for tracking the user's head, allowing users to look around in a simple virtual world, creating a strong sense of immersion.

His idea, it turned out, was two ideas. If the device was operated in a dark room, users would see only the world generated by the computer. In the 1980s, we would get to call this "virtual reality." If the lights in the room were on, however, users would see the virtual world overlaid onto the backdrop of the physical world—an interaction concept today known as "augmented reality."

Over the years, both ideas become highly influential in research. And while the two modes of his prototype looked indistinguishable, they turned out to afford quite different applications. VR systems tended to find application in simulation and ultimately also games. AR research, in contrast, tended to find use in providing users with timely information about their surroundings.

One of the key limitations of head-worn VR displays was they did not allow users to see their own hands and bodies anymore and no matter how much of a virtual world people wanted, they generally expected their bodies to be part of that world. On the one hand, this led to the development of systems that tracked the user's hands or even the entire body in order to provide a virtual one. On the other hand, the early 1990s saw the advent of the so-called CAVE system (that projects the virtual world onto the walls of a small room surrounding the user).

A few years later, AR got its own projection-based approaches. Initially projectors were used to project onto props (Raskar and colleagues), then also onto walls. While the CAVE was intentionally designed to be void of any proper structure or contents, stronger projectors allowed researchers and artists to overlay their projected contents onto real-world background objects materials. This approach became known as "Spatial Augmented Reality." As a result, AR applications diversified from information displays to large-scale artistic installations and media facades, blurring the distinction between VR and AR.

Today, almost 50 years after Sutherland's prototype, VR and AR are both becoming reality in the form of consumer devices, such as the VR headsets by Oculus and the Google Glass device. While Glass ships as a mere "heads-up" display, Microsoft's Hololens runs AR software on a collection of sensors including a depth camera that allows it to register its contents with the physical world.

Both devices are head-worn, but whether that ultimately is the way to go is subject to debate. The reason is that AR/VR are facing yet another requirement: as the computing world has become "social," VR and AR must follow. Consumers exploring virtual worlds will expect to be able to see their friends there. And while the promotion of head-worn VR displays by a social networking company suggests that "social" today may well mean "social with people located elsewhere," one might argue or at least hope that collocated friends and immediate social interaction will play a role in the future. Such collocated social use, however, clashes with head-worn VR displays, but even see-through glasses impact in part how people can perceive each other.

But if head-worn gear is not the solution, what will it be?

This is where the authors of IllumiRoom come in. They investigate what AR/VR could look like in a truly social environment—the living room. The project builds on an eight-year series of research projects in which the authors explore the boundaries between the virtual and the physical and I would encourage readers to read them all.

A key ingredient in the authors' approach to tackling AR/VR at home is the use of depth cameras, a type of device already available in the living room through Microsoft's Xbox gaming console. The camera is a key component, because it allows the authors to *integrate* their system with the living room. Rather than shutting out the room, as head-worn VR displays would, the authors digitize the living room and integrate it into the user's experience. They achieve all this with "living room-compatible" components. And rather than replace what they find in the living room, the authors augment it. They appropriate the TV and use an augmented reality approach to creating what might be considered more of a VR experience.

Ultimately, what impresses me about IllumiRoom is that it asks how virtual and real reality should come together. When I was a research scientist at Xerox PARC in the early 2000s, I was in the process of creating very large screens by combining flat panel displays with projection. I showed my prototype to the late Rich Gold, who liked the idea and agreed with my pitch about the productivity gains such big screens could provide. However, he was critical of my plan to simply maximize screen real estate. Bigger screens, he argued, were a trade-off: each piece allows users to access more virtuality, but at the expense of giving up a piece of the physical reality. And that is what the IllumiRoom authors excel at: creating a virtual reality while being mindful of the physical world. C

Patrick Baudisch (baudisch@acm.org) is a computer science professor at Hasso Plattner Institute, Potsdam, Germany.

Copyright held by author.

DOI:10.1145/2754391

IllumiRoom: Immersive Experiences Beyond the TV Screen

By Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson

Abstract

IllumiRoom is a proof-of-concept system that surrounds a television with projected light, bringing video games, and film experiences out of the TV screen and into the real world. IllumiRoom uses 3D scanning and projected light to change the appearance of the room, induce apparent motion, extend the field of view, and enable entirely new gaming experiences. Our system is entirely self-calibrating and is designed to work in any room. We present a detailed exploration of the design space of possible projected visualizations and we demonstrate ways to trigger and drive these visualizations from gaming content. We also contribute specific feedback from two groups of target users (10 gamers and 15 game designers); providing insights for enhancing game and film experiences beyond the TV.

1. INTRODUCTION

The television remains the focal point of living room entertainment today. While visual and audio quality has improved over the years, the content we watch (e.g., games, movies, TV shows) remains trapped inside our television screens. The resulting experiences are restricted by the physical size of the screen and ignore the surrounding physical environment.

In this paper, we propose a novel approach to enhance the viewing experience and blur the boundary between the on-screen content and the surrounding room (see Figure 1). We extend the visual experience outside of the television screen, using a projector that covers a wide area of the surrounding physical environment (e.g., your living room). Similar to Focus+Context displays,[3] the television provides a traditional, high-resolution gaming experience, and the projector provides low-resolution information for the user's peripheral vision. In contrast to previous work, we do not project onto a flat, white projection screen. Instead, we capture the appearance and geometry of the existing room (e.g., furniture, carpet, wallpaper) and use that information to create novel, interactive visual experiences.

In particular, our work investigates how projected visualizations in the periphery can negate, include or augment the physical environment, and thus enhance the content displayed on the television screen. We call such visualizations *peripheral projected illusions*. IllumiRoom uses these *illusions* to change the appearance of the room, induce apparent motion, extend the field of view (FOV), and enable entirely new virtual-physical gaming experiences. IllumiRoom is entirely self-calibrating, only requiring the user to point the projector at their TV, and it is designed to work in any room. The concept could be developed into a next generation game console with game content designed from the ground up, or it could be an "add-on" for existing consoles with gamers writing scripts to "mod" their favorite games. While our discussion focuses on interactive video games, the same or similar illusions could be used to enhance movies and television content.

2. MOTIVATING SCENARIO

To illustrate the idea behind IllumiRoom, imagine sitting down in your living room to play a video game on your television. When the game starts, the room magically transforms to look like a cartoon. The colors of the room become supersaturated and cartoon edges appear on your bookshelves, matching the shading in the video game. You come across an enemy in the game, and suddenly a bullet flies toward your character and then out of the television. The enemy throws a grenade toward you. The grenade rolls out of the television, bounces off your coffee table, and explodes in your living room, throwing shrapnel across your bookshelf. The entire living room appears to shake as you take damage from the explosion. This is just one of many experiences made possible with IllumiRoom.

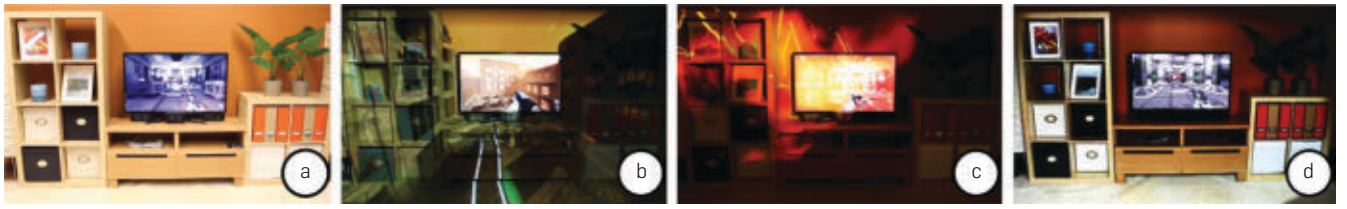
3. BACKGROUND

IllumiRoom uses a technology called *projection mapping* to bring gaming experiences out of the TV and into your living room. Normally, video projectors are used to display images on flat, white screens (e.g., PowerPoint presentations). Projection mapping points these same video projectors at non-flat objects, altering the appearance of everyday objects. Using animation and interactive rendering, everyday objects can be brought to life with projected light.[7, 18–20]

In academia, a number of projects have demonstrated the use of projectors to augment workspace environments,[16, 18, 24]

The original version of this paper is entitled "IllumiRoom: Peripheral Projected Illusions for Interactive Experiences" and was published in the *Proceedings of the 2013 ACM SIGCHI Conference on Human Factors in Computing Systems*.

Figure 1. IllumiRoom is a proof-of-concept system that augments the physical environment surrounding a television to enhance interactive experiences. We explore the design space of projected visualizations that can negate, include, or augment the surrounding physical environment. (a) With a 3D scan of the physical environment we can (b) directly extend the FOV of the game, (c) selectively render scene elements, (d) augment the appearance of the physical environment, here as a cartoon. See the video online at: http://youtu.be/L2w-XqW7bF4.



and even gaming experiences.[4,11,24] Previous work has explored ways to adapt to non-white projection surfaces, through a process called radiometric compensation,[19] for example, turning a curtain or a building facade into an almost ideal projection surface (e.g., Bimber et al.).[5]

In the last decade, projection mapping has exploded in popularity,[12] being used for outdoor advertising on buildings, special effects in live venue entertainment, and in artistic installations. Currently, artists can create amazing effects by rendering a video that is specific to the scene and projector location. Inspired by these art pieces, IllumiRoom generates real-time, interactive illusions that automatically adapt to any room, any television and any game.

3.1. Displaying in the periphery

Large, high-resolution displays can be expensive. To create an immersive display at a lower cost, Baudisch et al. proposed *Focus+Context* (F+C) displays,[3] which have a high-resolution computer monitor (focus) surrounded by a lower resolution projection screen (context). This display configuration matches the setup of the human visual system. Compared to the center of gaze, peripheral vision is highly sensitive to motion, but is relatively poor at distinguishing color and shape.[21] This is due to the lower density of receptors and the changing distribution of receptor types outside the fovea. Peripheral vision plays an important role in the perception of motion[10] and increases feelings of immersion.[22] In IllumiRoom, the smaller display serves the high-resolution capability of the fovea, while the projector serves the low-resolution, motion sensitive periphery.

The stimulation of peripheral vision could alternatively be addressed by a single, very large display, a F+C screen,[3] a CAVE setup,[8,15] or even low-resolution LEDs arrays[9] (e.g., Phillips Ambilight TV).[23] Along these lines, a recent project by the BBC explored F+C passive film experiences using a projector and a television,[14] but did not have any 3D scene knowledge or interactivity. IllumiRoom advances these ideas to high-resolution projected content which is dynamically adapted to the appearance and geometry of the room. Rather than relying on separate projection screens, IllumiRoom adapts the projections to the user's existing living room, and demonstrates a large range of possible peripheral projected illusions.

4. ILLUMIROOM SYSTEM

Our vision for a fully developed IllumiRoom system includes an ultra-wide FOV device sitting on the user's coffee table, projecting over a large area surrounding the television (see Figure 2a). The device would be connected wirelessly to a next generation gaming console as a secondary display. The ultra-wide FOV could be obtained with an ultra-short throw projector, or by coupling a standard projector with a spherical/parabolic mirror (as the peripheral content does not need a uniform pixel aspect ratio). The room geometry could be acquired with a depth sensor or a structured light scan (which would require only a web camera).[2,5,11]

Our current proof-of-concept prototype uses a commodity wide FOV projector (InFocus IN126ST) and a Microsoft Kinect for Windows sensor (see Figure 2b). The prototype is restricted by the FOV of the Kinect and projector (57° horizontal FOV). Therefore, the system is mounted above and behind the user's head, as they are seated on a couch in front of the television.

The Kinect sensor captures the color and geometry of the scene, which is used during calibration and to render the peripheral projected illusions that respond to the living room furniture in a physically realistic manner (similar to the ideas in Benko et al.,[4] Wilson et al.[24]). Careful calibration of the system is required in order for the illusions to tightly match the on-screen content and the physical environment. The calibration of the IllumiRoom system is fully automatic, determining the relative pose of the projector to the depth sensor and the position of the television in the projector image. Therefore, setup only requires that the projector and depth camera are placed such that they cover the area surrounding the TV.

The automatic calibration projects Gray code sequences[2] to establish dense correspondences between the projector and the Kinect's color camera. Using the default parameters for the Kinect, we transform these correspondences into 3D points. We then solve for the intrinsic and extrinsic parameters of the projector. As there is no depth data for specular surfaces, such as that of shiny, black televisions, we recover the position of the TV using a 2D homography, with virtual markers displayed in each corner of the screen. It is important to note that the calibration need only happen once. IllumiRoom could easily be extended to continuously capture the geometry, enabling ad hoc changes to the living room such as moving furniture and people.

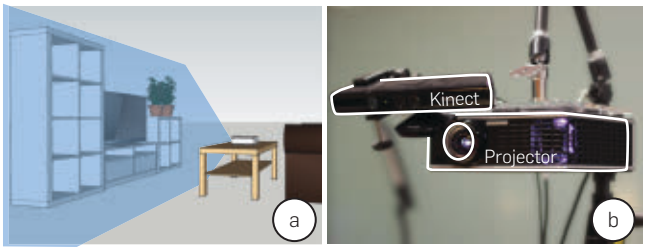
4.1. Radiometric compensation

IllumiRoom projects on the existing living room furniture, and therefore cannot rely on the presence of a white, flat projection surface. Using radiometric compensation[5,19] the projected light may be modulated to achieve a desired color, given knowledge of the existing surface color and geometry. However, this process is limited by the brightness, dynamic range, and color primaries of the projector, as well as the existing surface color, material, and geometry. Accordingly, some desired surface colors are un-achievable. For example, a saturated red object in our physical environment cannot be made to appear a saturated green color with our low cost commodity projector.

Therefore, all of the illusions use the fact that they occur in peripheral vision. Illusions that may not seem realistic on close examination may be quite effective in the user's peripheral vision. Most illusions do not attempt to accurately reproduce a scene, but aim to introduce motion in the periphery. Also, subtly modifying the existing surface color may be more effective than replacing the color entirely.

Illusions that modify the surface color and not the apparent geometry of the room are independent of viewing position, and thus can be viewed by multiple users from any position within the room. Illusions that add to or modify the geometry of the room are inherently dependent on viewing position and are most effective when viewed from a fixed point within the room. Generally, as the room's furniture becomes more complex, effects which virtually modify the geometry are more sensitive to viewing position.

Figure 2. Our vision for a productized IllumiRoom system includes an ultra-wide field of view device that sits on a coffee table (a) and projects content in the area surrounding the television. (b) Our current proof-of-concept prototype uses an off-the-shelf projector and a Kinect sensor.



5. ILLUMIROOM ILLUSIONS

We envision the IllumiRoom system supporting a great variety of peripheral projected illusions, limited only by the imagination of game designers. We have prototyped illusions that we believe represent the primary dimensions of the design space (discussed in the next section). This represents an initial survey of the design space and is not exhaustive. We outline the techniques used to create the illusions, and describe the illusions that were implemented. The illusions are best understood by demonstration (see the video online at: http://youtu.be/L2w-XqW7bF4).

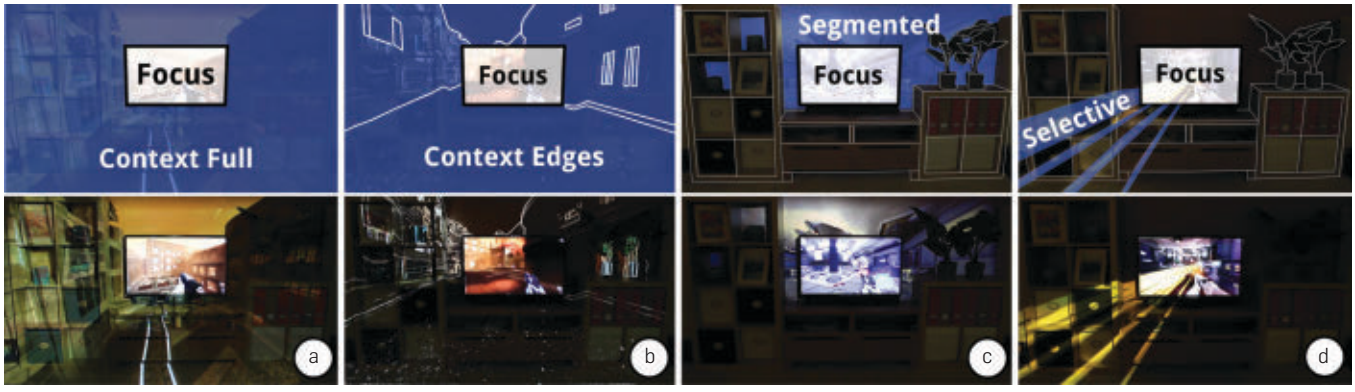
We present the illusions as distinct points in the design space. However, most of the illusions can be combined together, forming a rich and flexible palette for game designers.

5.1. Focus+Context

The most obvious way to increase the user's sense of immersion is to extend the content from the television screen out into the room, replacing the physical reality with the game's reality. We call this *F+C Full* (see Figure 3a). This is essentially Focus+Context displays,[3] but using a non-flat, non-white projection surface, with radiometric compensation. As stated above, the ability to compensate for the existing surface color is limited, and so this effect relies on the fact that the user will directly attend to the television screen rather than the periphery.

The next illusion, *F+C Edges*, provides a maximum contrast representation of the peripheral content by displaying only the black-and-white edge information (Figure 3b). This illusion is more robust to ambient

Figure 3. Projecting wider field of view game content around the TV can increase immersion. (a) *F+C Full*: The full game content is projected with radiometric compensation onto the surrounding furniture. (b) *F+C Edges*: A high-contrast version where only the edges in the game content are projected. (c) *F+C Segmented*: Game content is projected only onto the flat, rear wall. (d) *F+C Selective*: Selective game content (e.g., bullets) bleed out of the TV.



light in the room, and increases peripheral optical flow.

F+C Segmented extends the game only onto the rear wall surrounding the television (Figure 3c). While the furniture in the room may not be ideal for projection, most televisions are at least partially surrounded by a flat, light colored wall which provides an ideal, view independent projection surface. This rear wall could be found by a recursive RANSAC plane fitting procedure,[17] seeded with the location of the television. Our prototype currently relies on a manually specified image mask identifying the rear wall.

While the previous illusions bleed all game information into the physical environment, there are times when a more subtle approach may be appropriate. For example, when watching a horror movie, an atmosphere of immersion and suspense is created by the absence of the villain, and then their sudden appearance. When all information is present all the time, there is no opportunity for suspense or surprise. With *F+C Selective* only certain game elements escape the television (Figure 3d). For instance, a first-person shooter might bleed only weapons fire or explosions out of the television. Or, markers representing other characters or key items in the game may be selectively displayed, thus increasing their emphasis.

These illusions increase immersion, induce apparent motion, and provide additional information about the game content. The F+C illusions all require access to the video game's rendering process.

5.2. Peripheral Flow

If we do not have access to the rendering process, apparent motion may still be induced through peripheral visual flow, as long as we have information about the motion of the game's camera. In *Grid* (Figure 4a), an infinite grid moves with the motion in the game. When the user moves or looks left or right, the grid moves in the opposite direction. When the user moves forwards or backwards, the grid zooms in or out.

Similarly, *Starfield* (Figure 4b) creates peripheral flow by moving 3D points to match the movement of the game's camera. Points are randomly generated in the virtual space between the TV and the user, according to the position and orientation of the game's camera.

5.3. Color augmentation

The physical environment may be augmented to match the theme or mood of the game. For instance, the room colors may be super-saturated to resemble a cartoon by simply projecting the color of the surface back onto itself,[6] and then drawing silhouette edges black (Figure 5). Similarly colors may be desaturated to create a black-and-white, "film noir" look. We leave more complex texture replacement techniques (i.e., procedural texturing) for future work.

5.4. Texture displacement

Imagine an explosion in a video game or film that is so powerful that it shakes your entire room. IllumiRoom can use projected light to modify the appearance of objects in the room, making them appear to move, change size, or distort in more extreme ways. Due to the projector's limited ability to compensate for surface color, these illusions are most effective if the displacements are small and short-lived. *Radial Wobble* (Figure 6) distorts the apparent room geometry in a "force field" ripple effect. The illusion displaces the original surface texture in an expanding, radial sinusoidal wave.

5.5. Lighting

With full control over the illumination, the lighting in the physical environment may be altered.[19,20] Room lighting can change based on the mood or theme in the game. For example, the look of a scene set in space might be achieved by illuminating with point light sources and harsh shadows. The *Lighting* effect attempts to match the lighting conditions of the physical environment with the virtual environment (Figure 7). Our prototype implements soft shadows

Figure 4. Apparent motion may be induced without access to a game's rendering by displaying abstract peripheral visual flow. (a) An infinite *Grid* moves with the game's camera. (b) A 3D *Starfield* of spheres moves with the game's camera.

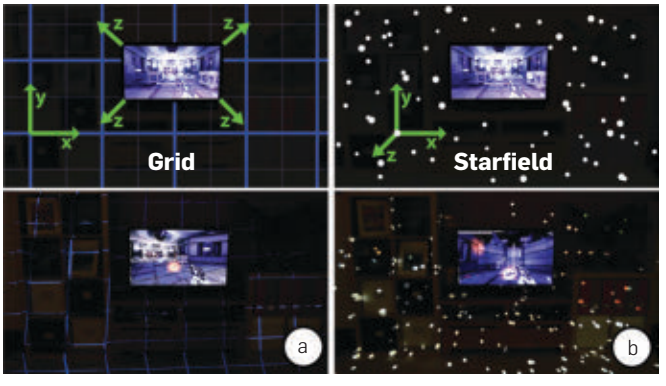


Figure 5. *Appearance*: The appearance of the physical environment may be augmented (a), to match the theme of the game. (b) Cartoon super-saturated colors and silhouette edges. (c) Black-and-white (desaturated) colors.



and a single point-light source, matching the closest point-light source in the game

5.6. Physical interactions

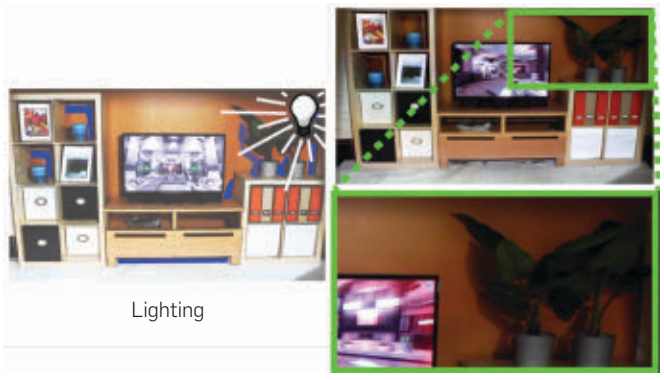
In F+C Selective elements from the game (e.g., bullets) break the screen barrier and fly into the physical environment; however, there are no physical interactions with the environment. *Bounce* envisions an element in the game leaving the virtual environment and interacting with physical reality. In our prototype, a grenade rolls out of the television, and then bounces and rolls around the physical environment according to a physics simulation using the room geometry (Figure 8a). From a design perspective, it is important to note that the designer does not know where the grenade will go after it leaves the TV, as it will adapt to the 3D geometry of the user's room.

In *Snow*, falling snow interacts with the physical environment, briefly collecting on surfaces in the room. Similar to *Starfield*, the snow moves according to the movement of the user in the virtual environment, allowing the user to walk or drive through snow (Figure 8b). Our prototype simulates the accumulation of snow by turning surfaces with an upward facing normal gradually white.

Figure 6. *Radial Wobble*: To create a "force field" ripple effect, the physical surface texture is displaced with a radial expanding sinusoidal pattern.



Figure 7. *Lighting*: The lighting and shadows of the physical environment are matched to lighting in the virtual environment.



6. DESIGN SPACE

The range of possible peripheral projected illusions is as great as the range of visual effects in video games or film. In order to better understand the full range of illusions, we identify the primary dimensions of the design space. Illusions are defined by three primary factors: the goals of the illusion, the connection or separation from physical reality, and the level of abstraction from the game content.

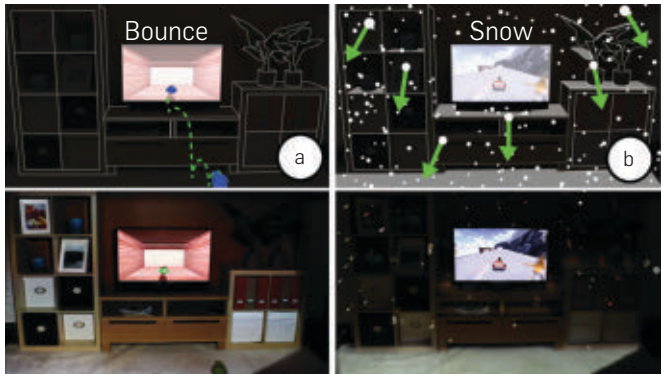
6.1. Illusion goals

Peripheral projected illusions can enhance gaming experiences in a variety of ways. All of the proposed illusions attempt to increase the user's sense of immersion by increasing the FOV. All of the illusions increase the user's sense of presence, making the user feel "in the game." Additionally, the illusions can induce apparent motion, provide additional information and content, create a sense of atmosphere or theme, and support entirely new physical–virtual game mechanics. These goals are not mutually exclusive, and a single illusion may support multiple goals.

6.2. Connection to reality

IllumiRoom enables designers to explore the entire spectrum of the reality-virtuality continuum;[13] by negating, including, or augmenting the surrounding physical environment. Designers can negate the physical environment, making the living room disappear; thereby immersing the user in an entirely virtual environment (e.g., *F+C Full, F+C Edges, Grid, Starfield*). Or the game can be grounded in reality, by including the physical environment and selectively introducing virtual objects into physical reality (e.g., *F+C Seg, F+C Sel, Bounce, Snow*). Finally, IllumiRoom enables a unique middle ground, where the physical surroundings are augmented to match the virtual environment, creating a mixed reality where it is not clear what is real or virtual (e.g., *Appearance, Lighting, Radial Wobble*).

Figure 8. Virtual objects can realistically interact with physical objects. (a) In *Bounce* a grenade rolls out of the TV and bounces around the living room. (b) In *Snow*, falling snow moves according to the user's movement in the game and interacts with the physical environment.



6.3. Abstraction

The illusions can be dependent or independent of the underlying game content. For instance, if we extend the exact game content outside of the TV, the illusion is entirely dependent on the game content (e.g., *F+C Full, F+C Edges, F+C Seg, F+C Sel, Bounce*). However, if we display an abstract visualization, such as a moving grid, then the peripheral illusion is independent of the game content (e.g., *Grid, Starfield, Snow, Appearance, Lighting*, and *Radial Wobble*).

7. TRIGGERS

The illusions can be connected with game content through a variety of means. Ideally, IllumiRoom would be directly integrated into a next generation console and new games would be designed for IllumiRoom from the ground up. We envision an API that enables triggering illusions, changing surface appearance, controlling room lighting, inserting objects into the physical environment, etc. In our prototype, to make the system work with numerous games, the IllumiRoom system is a separate component from the game examples, and networking messages are sent to share information.

If the source code of the game is available, then the game can be easily modified to trigger effects. Game events can trigger illusions to be toggled on and off via simple network messages. The position and orientation of the player in the virtual world can be sent to drive the Peripheral Flow illusions. The game can support F+C illusions by conducting a second rendering pass using a wider FOV. In our prototype this information is sent to the IllumiRoom system through a texture residing in shared memory.

While the source-code may not be easily available for most games, many of the illusions can be controlled by other means. For example, the Peripheral Flow illusions need only some measurement of the player/camera movement. An approximation can be obtained by real-time image analysis of the rendered game (e.g., optical flow). In our prototype, any game's image may be captured using a video capture card (DeckLink Intensity Pro), and subsequently analyzed by optical flow techniques to drive the illusions.

A significant amount of game state may be inferred solely from the user's input through the controller. Given knowledge of the game and the controller mapping, controller input may be mapped directly onto illusions. For instance, if the game is a first-person shooter and the player presses the "Right Trigger" button on the controller, then the player must be shooting and the *Radial Wobble* illusion may be activated. In our prototype, input from the controller and vibrotacticle output sent to the controller from the game can drive illusions. This works with any game. We envision users scripting and sharing "mods" of their favorite games to work with the IllumiRoom system.

For our prototype, games are run on the same PC as the IllumiRoom system. GlovePIE intercepts and sends game controller input to the IllumiRoom system by networking messages. Vibrotactile output at the controller is captured by using an Xbox controller modified with an Arduino.

8. USER EVALUATIONS

We evaluated the IllumiRoom concept through two user studies focused on end-user "gamers" and expert game designers respectively. In both user studies, we were primarily interested in determining user perceptions along different dimensions of the design space. Just as there is no "best visual effect" for film or video games, we were not expecting to identify a "best illusion." Rather, we sought the strengths and weaknesses of each technique. We summarize the results here and refer the reader to the full paper for details.

8.1. End-user feedback

We recruited 10 participants (ages 20–30, two females), who were familiar playing first-person shooter games with an Xbox controller. Each user interacted with our 11 illusions, which we paired with matching game content to create complete immersive experiences. We used a variety of open source games: Red Eclipse, SuperTuxKart, Unity3D's race car example, and a custom-built DirectX application for *Bounce*. See the video (http://youtu.be/L2w-XqW7bF4) for the complete experiences.

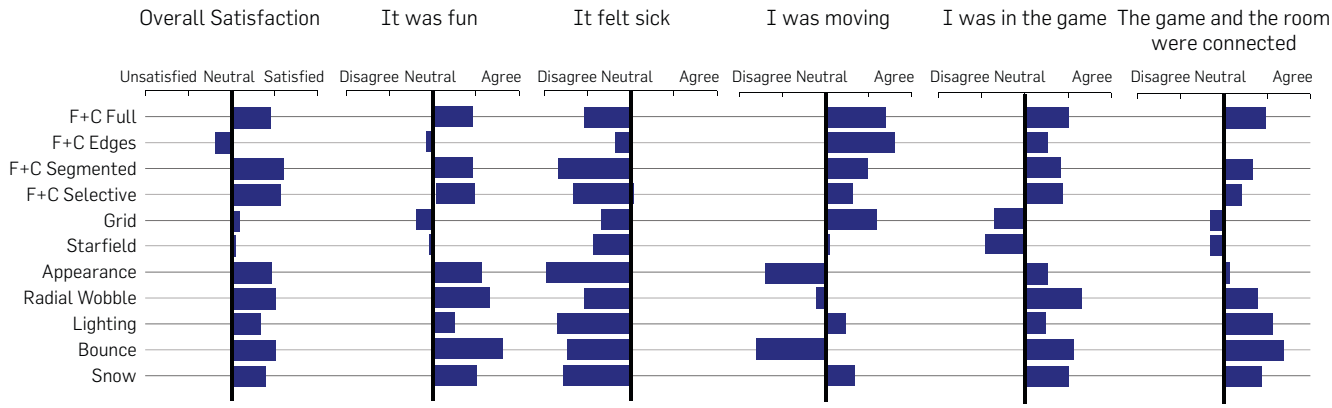
For each illusion, users first interacted with the game content only on the TV, then the illusion was activated. After viewing each illusion in a randomized order, participants rated and ranked the illusions on various dimensions using a card sorting task. First rating on "overall satisfaction," then in response to the following prompts "It was fun," "I felt sick," "I felt like I was moving," "I felt like I was in the game world," and "The game and the physical world felt like separate spaces."

Participants responded very positively to the illusions and the concept of IllumiRoom, see Figure 9. As expected, there was not one "winning" illusion, but a diverse set of illusions that have strengths on different dimensions of the design space. Participants found the F+C illusions some of the most useful for game play, because they naturally provide additional gaming information. For instance, one user commented that *F+C Selective* was "very useful for finding items, it kinda lets you see just pieces of the game that are important to know around you."

However, the F+C illusions can also make the user overwhelmed or more prone to motion sickness. One user commented that with *F+C Full*, "There was just a lot going on. To see everything going on around you it was like really cool but especially in as fast-paced a game as that, it was just too much happening at once." The *F+C Selective* and *F+C Segmented* illusions seem to represent a middle ground between immersion, adding content information and balancing user comfort. As one user said about *F+C Selective*,

http://glovepie.org/glovepie.php.
http://redeclipse.net.
http://supertuxkart.sourceforge.net.
http://unity3d.com/.

Figure 9. Mean rating for each illusion evaluated in the user study. Participants rated "overall satisfaction" and responded to five questions on a five-point Likert scale from "strongly disagree" to "strongly agree."



"Out of all of these that would be the most useful and immersive without being distracting." Similarly, for *Grid* and *Starfield*, multiple users commented that they had a greater sense of motion, but missed the contextual information of the F+C illusions.

Appearance worked well at creating a sense of atmosphere for the game. As one user put it "The appearance is awesome. Like that could just be an effect in someone's living room all the time. That is just super cool." *Radial Wobble* was another "magical" effect, that "kinda made you feel like you are physically doing something in [the game]." However, some users said the effect was off-putting, because "all your stuff is shaking in the real world." As one user stated, "It just seemed like one that you should save for really impactful moments. So I just wouldn't want it all the time."

Lighting created a better sense of immersion and presence for some users. "I think more so than any of the others, that one made me feel like I was in the game world." For *Bounce*, multiple users referenced the grenade indicator in games like Infinity Ward's Call of Duty, "Most of the time I don't see the grenades at all . . . They always kill me, so that was a good one." The *Snow* illusion created a sense of atmosphere and movement for users; however most users did not notice the physical interactions with the geometry. This is because the snow quickly "melted" after impact. One user commented that they "really wanted it to pile up more . . . if it was piling up it would be an indication of how long I was playing and that would be cool."

8.2. Game designer feedback

In addition to gamers, we also elicited informal feedback about our illusions from game designers ($N = 15$). We showed the illusions to three groups, each composed of five game designers or game artists. Each participant had more than five years of experience in the game industry.

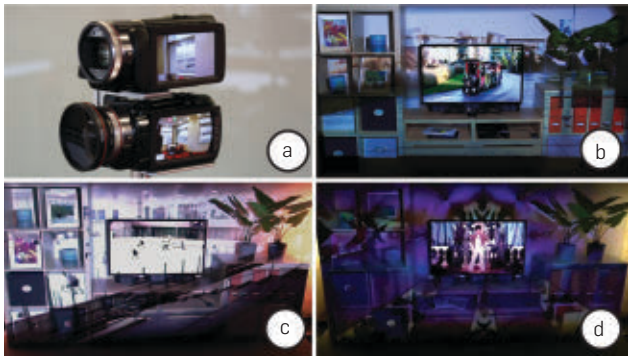
While individual game designers had different preferences, we extracted the commonalities among designers. Generally, the game designers thought that *F+C Full* was impressive at first, but "would probably not hold up to the long term play." They thought the most promising illusions were ones that altered the mood (*Appearance*), altered the players sense of reality (*Radial Wobble*), or ones that selectively showed game content without becoming overwhelming (*F+C Selective, F+C Segmented, Bounce*). They expressed concern over illusions that need a tightly, fine-tuned connection with the game (*Lighting*); as they were worried about destroying the illusion for the user if the illusion was not perfectly matched to the game content. Therefore, they saw the merits of *Snow* and *Starfield*, which have much looser connections with the source content. They all suggested games where IllumiRoom could be applied (e.g., Forza, Skyrim, Halo, Portal) and enthusiastically suggested that they would like to have a setup like this at home.

8.3. Extending film and video

While our discussion has focused on interactive video games, IllumiRoom can also extend the viewing experience of film and television. This panoramic video content can be displayed around the television using the *F+C* techniques (e.g., *F+C Full*). For example, Figure 10c shows a hockey game that we recorded, where the "focus" video is tightly held on the players, capturing their actions, while the "surround" video is immersing the viewer with live panoramic views of the stadium and the fans at the game. Or in Figure 10d, we demonstrate a live fashion show where the "focus" video captures the models on the runway and the "surround" video features abstract visuals that match the theme of the outfits.

To begin experimenting with such video experiences, we built a custom dual-camera rig which consists of two standard 1080p video cameras: a narrow FOV "focus" camera and a wide FOV "surround" camera using a wide angle lens. Alternatively, IllumiRoom video content could be captured with a ultra high resolution (4K) camera, or traditional video content could be extrapolated (e.g., Aides et al.,[1] Novy and Bove).[15] Our experience with recording and watching such dual feed videos in IllumiRoom leads us to believe there are a whole new set of cinematic effects that are possible by

Figure 10. IllumiRoom can also be used for panoramic film and video. (a) A custom dual-camera rig for capturing F+C video (a). The panoramic video content can be displayed on top of the existing living room furniture (b). Sports games can be directly extended providing additional context (c). More abstract content can be used in the periphery like these abstract visualizations surrounding a fashion runway video (d).



creatively combining the two video feeds.

9. CONCLUSION

In this paper, we presented a proof-of-concept system for augmenting the physical environment surrounding a television to create a more magical, immersive experience. We presented 11 peripheral projected illusions that demonstrate the variety of the design space. We then elicited feedback from gamers and game designers about the illusions along dimensions of the design space. While we did not formally compare the experience of the IllumiRoom system with a standard TV setup, the very positive feedback of both user groups indicates that there is great promise in the approach.

We explored 11 points in the design space, but there are many peripheral projected illusions left to be explored. We only scratched the surface of what is possible with a shared physics representation between the virtual and physical environments (e.g., *Bounce*). Also, while we examined each illusion individually, future work should examine the combination of illusions and how this palette of illusions could be used by game designers. As some of our illusions distort reality, future work should explore how far users are willing to push their concept of reality, in their own living room.

Another promising direction is to further explore illusions specifically for video content (e.g., TV shows or full-length feature films). Can a grenade from the latest Bond film explode in your living room? How would such content be authored? It would be important to investigate how the movie director should deal with the fixed nature of a film and the randomness imbued by the real-time system adapting to the user's living room.

Additionally, our methods for connecting content and illusions are not exhaustive. It should be possible to use other cues (e.g., audio) to trigger illusions. For instance, a loud explosion in a film could trigger a *Radial Wobble* illusion. Simple techniques such as these could enable users to "mod" existing films and games, and share the scripts.

Finally, before the IllumiRoom system can be in every living room, the final form factor, cost, and computational requirements of the system must be determined. While there are many unanswered questions about the future of gaming and film beyond the TV, we hope we have demonstrated that they are worth answering. C

References

1. Aides, A., Avraham, T., Schechner, Y. Multiscale ultrawide foveated video extrapolation. In *IEEE ICCP*, 2011.
2. Battle, J., Mouaddib, E., Salvi, J. Recent progress in coded structured light as a technique to solve the correspondence problem: A survey. *Pattern Recognition 31*, 7 (1998), 963–982.
3. Baudisch, P., Good, N., Stewart, P. Focus plus context screens: Combining display technology with visualization techniques. In *ACM UIST*, 2001.
4. Benko, H., Jota, R., Wilson, A.D. MirageTable: Freehand interaction on a projected augmented reality tabletop. In *ACM CHI*, 2012.
5. Bimber, O., Emmerling, A., Klemmer, T. Embedded entertainment with smart projectors. *IEEE Computer 38*, 1 (Jan. 2005), 16–26.
6. Bimber, O., Iwai, D. Superimposing dynamic range. *ACM Trans. Graphic 27*, 5 (Dec. 2008), 1.
7. Bimber, O., Raskar, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. AK Peters, 2005.
8. Cruz-Neira, C., Sandin, D., DeFanti, T. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *ACM SIGGRAPH*, 1993.
9. Ghosh, A., Trentacoste, M., Seetzen, H., Heidrich, W. Real illumination from virtual environments. In *Eurographics Symposium on Rendering*, 2005.
10. Gibson, J. *The Perception of the Visual World*. 1950.
11. Jones, B., Sodhi, R., Campbell, R., Garnett, G., Bailey, B.. Build your world and play in it: Interacting with surface particles on complex objects. In *IEEE ISMAR*, 2010.
12. Jones, B., Sodhi, R., Karsch, K., Casperson, M. Projection mapping central, 2014. http://projection-mapping.org.
13. Milgram, P., Takemura, H., Utsumi, A., Kishino, F. Augmented reality: A class of displays on the reality-virtuality continuum. *Proc. SPIE 2351, Telemanipulator Telepresence Technol. 282*, (1995).
14. Mills, P., Sheikh, A., Thomas, G., Debenham, P. Surround video. In *Networked and Electronic Media Summit*, 2011.
15. Novy, D.E., Bove, V.M. Infinity-by-nine, 2013. http://obm.media.mit.edu.
16. Pinhanez, C. The everywhere displays projector: A device to create ubiquitous graphical interfaces. In *ACM UbiComp*, 2001.
17. Quirk, P., Johnson, T., Skarbez, R., Towles, H., Gyarfas, F., Fuchs, H. RANSAC-assisted display model reconstruction for projective display. In *IEEE VR*, 2006.
18. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., Fuchs, H. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *ACM SIGGRAPH*, 1998.
19. Raskar, R., Welch, G., Low, K.K., Bandyopadhyay, D. Shader lamps: Animating real objects with image-based illumination. In *Eurographics Workshop on Rendering Techniques*, 2001.
20. Raskar, R., Ziegler, R., Willwacher, T. Cartoon dioramas in motion. In *ACM NPAR*, 2002.
21. Sekular, R., Blake, R. *Perception*, 2nd edn. 1990.
22. Tan, D.S., Gergle, D., Scupelli, P., Pausch, R. With similar visual angles, larger displays improve spatial performance. In *ACM CHI*, 2003.
23. Weffers-Albu, A., de Waele, S., Hoogenstraaten, W., Kwisthout, C. Immersive TV viewing with advanced Ambilight. In *IEEE ICCE*, Jan. 2011.
24. Wilson, A.D., Benko, H., Izadi, S., Hilliges, O. Steerable augmented reality with the Beamatron. In *ACM UIST*, 2012.

Brett R. Jones (brettjonesr@gmail.com), University of Illinois at Urbana-Champaign, Urbana, IL.

Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson ({benko, eyalofek, awilson}@microsoft.com), Microsoft Research, Redmond, WA.

Watch the authors discuss this work in this exclusive *Communications* video.

Copyright held by author. Publication rights licensed to ACM. $15.00.

Army Research Laboratory
Multiple Positions

Title:
Network Science Research Scientist

The **United States Army Research Laboratory** seeks scientists for post-doctoral and full-time research positions in the area of Network Science, focusing on theoretical and experimental aspects of quantification of networks' structure, behaviors; dynamics and evolution of networks; and emergent and self-organizing networks and their behavior. Areas of particular interest include experimental approaches to research in multi-genre (socio/cognitive, informational and communication) networks to include network theory, architectures, and experimental methods, and research in tactical wireless networking combining traditional communications networking techniques with content based software defined networking. The research requires expertise in some of the following: Network Science, social networks, network theory, wireless communications, and software defined networking.

Open to U.S. citizens. At a minimum, applicants should have a Bachelor's degree in Computer Science, Electrical Engineering, Mathematics or related field. A Master's or PhD is highly desired.

The positions are located in Adelphi, Maryland, north of Washington, DC.

Send a cover letter and resume/CV to Dr. Brian Rivera, brian.m.rivera.civ@mail.mil, to initiate the application process. Contact number: (301) 394-2298.

Title:
Cyber Security Research Scientist/Engineer

The United States Army Research Laboratory seeks scientists and engineers for post-doctoral and full-time research and development positions in the area of Cyber Security, focusing on theoretical and applied aspects of characterizing system and network vulnerabilities and risk analysis; automation of system vulnerability assessments; automated analysis of patterns in malicious activities; methods for detection of anomalies, network intrusion, attack, and data ex-filtration; methodologies for experimental validation of network security tools; cyber forensics; threat behavior analysis, security of supervisory control and data acquisition systems, computer network defense metrics, continuous monitoring, supply chain, fingerprinting; and big data analytics for cyber security.

Open to U.S. citizens. At a minimum, applicants should have a Bachelor's degree in Computer Science, Electrical Engineering, Mathematics or related field. A Master's or PhD is highly desired.

The positions are located in Adelphi, Maryland, north of Washington, DC, and at Aberdeen, Maryland, north of Baltimore.

Send a cover letter and resume/CV to Dr. Alexander Kott, alexander.kott1.civ@mail.mil , to initiate the application process. Contact number: (301) 394-1507.

Title:
Quantum Information Science: Information Teleportation Networks Research Scientist

The United States Army Research Laboratory (ARL) seeks scientists for post-doctoral and full-time research positions in the area of structures, behaviors, and functionality of quantum networks including theories and models aimed at understanding quantum-network phenomena, network protocols, algorithms and architectures; techniques and methods that enable the design and characterization of quantum networks for Army-relevant operations that are unattainable by classical systems. Research includes network topologies and protocols and associated algorithms, and architectures that optimize overall network properties, e.g., quantum capacity; dynamics and transient behavior of a multi-user heterogeneous quantum/classical networks, including mobile, free-space ones; managing protocols for efficient entanglement routing, control and manipulation; defining and characterizing network metrics. Research also includes fundamental properties of communication channels enabling quantum communication; loss and decoherence in quantum state transfer and entanglement distribution over channel of various physical implementations; fundamental and technological performance limitations, theories and models for determining trade-offs imposed by various network modules and architectures.

Open to U.S. citizens. At a minimum, applicants should have a Bachelor's degree in Physics or related field. A Master's or PhD is highly desired.

The position is located in Adelphi, Maryland, north of Washington, DC.

Send a cover letter and resume/CV to Dr. Brian Rivera, brian.m.rivera.civ@mail.mil, to initiate the application process. Contact number: (301) 394-2298.



上海科技大学
ShanghaiTech University

TENURE-TRACK AND TENURED FACULTY POSITIONS IN INFORMATION SCIENCE AND TECHNOLOGY

The newly launched **ShanghaiTech** University invites talented faculty candidates to fill multiple tenure-track/tenured positions as its core founding team in the School of Information Science and Technology (SIST). Candidates should have outstanding academic records or demonstrate strong potential in cutting-edge research areas of information science and technology. They must be fluent in English. Overseas academic training is highly desired. Besides establishing and maintaining a world-class research profile, faculty candidates are also expected to contribute substantially to graduate and undergraduate education within the school.

ShanghaiTech is matching towards a world-class research university as a hub for training future generations of scientists, entrepreneurs, and technological leaders. Located in a brand new campus in Zhangjiang High-Tech Park of the cosmopolitan Shanghai, ShanghaiTech is at the forefront of modern education reform in China.

Academic Disciplines: We seek candidates in all cutting edge areas of information science and technology that include, but not limited to: computer architecture and technologies, micro-electronics, high speed and RF circuits, intelligent and integrated information processing systems, computations, foundation and applications of big data, visualization, computer vision, bio-computing, smart energy/power devices and systems, next-generation networking, statistical analysis as well as inter-disciplinary areas involving information science and technology.

Compensation and Benefits: Salary and startup funds are internationally competitive, commensurate with experience and academic accomplishment. We also offer a comprehensive benefit package to employees and eligible dependents, including housing benefits. All regular faculty members will be within our new tenure-track system commensurate with international practice for performance evaluation and promotion.

Qualifications:
- Ph.D. (Electrical Engineering, Computer Engineering, Computer Science, or related field)
- A minimum relevant research experience of 4 years.

Applications: Submit (in English, PDF version) a cover letter, a 2-3 page detailed research plan, a CV with demonstrated strong record/potentials; plus copies of 3 most significant publications, and names of three referees to: sist@shanghaitech.edu.cn. For more information, visit http://www.shanghaitech.edu.cn.

Deadline: June 30, 2015 (or until positions are filled).

ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to *acmmediasales@acm.org*. Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.

Estimates: An insertion order will then be e-mailed back to you. The ad will by typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.

Rates: $325.00 for six lines of text, 40 characters per line. $32.50 for each additional line after the first six. The MINIMUM is six lines.

Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact:
acmmediasales@acm.org

Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at:
http://jobs.acm.org

Ads are listed for a period of 30 days.

For More Information Contact:
ACM Media Sales
at 212-626-0686 or
acmmediasales@acm.org

Bowdoin College
Visiting Assistant Professor

The Computer Science Department at Bowdoin College seeks a one-year, full-time Visiting Assistant Professor, to begin July 1, 2015. Four course teaching load. Expertise in bioinformatics, graphics and visualization, machine learning, security, computer architecture, databases, or high performance computing preferred. We seek applicants who can benefit from and contribute to Bowdoin's effort to build an environment that supports computational approaches across all disciplines. Ph.D. at time of appointment required.

Visit https://careers.bowdoin.edu to apply. Review of applications begins March 23. Bowdoin College is committed to equality through Affirmative Action, and is an equal opportunity employer.

The University of Texas at Tyler
Computer Information Systems
College of Business and Technology
Assistant Professor & Senior Assistant/ Associate Professor

Applications are invited for two positions in the Department of Computer Science and Computer Information Systems in the College of Business and Technology at the University of Texas at Tyler beginning August 2015. One position is for an Assistant Professor and the other is for a Senior Assistant/Associate Professor.

Requirements include a Ph.D. in computer information systems, demonstrated research creativity and productivity, ability to develop a strong research program, evidence of effective teaching and other communication skills. For an appointment at the Associate level, the successful candidate must have an established research and teaching portfolio.

Please send (**as a single PDF file**) cover letter, detailed curriculum vitae, and names and contacts of at least three references to Katherine Garibaldi at kgaribaldi@uttyler.edu.

Review of applications will begin May 1, 2015 and continue until a suitable candidate is found.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Faculty Positions in Digital Humanities
at the Ecole polytechnique fédérale de Lausanne (EPFL)

The College of Humanities at EPFL invites applications for junior and senior faculty positions within its newly created Digital Humanities Institute. We seek individuals who will develop and drive an interdisciplinary research program at the intersection of the humanities and computer science/engineering, who are dedicated to teaching at the undergraduate and graduate levels, and who will be active members of the growing digital humanities community at EPFL. Candidates from all areas that use computational methods in humanities research will be considered, but preference will be given to candidates in the fields of art history, museology, musicology, big data for the humanities, and information visualization.

EPFL offers a fertile environment for research cooperation between different disciplines including the humanities. It hosts the archives of the Montreux Jazz Festival (http://metamedia.epfl.ch), is involved in a trans-disciplinary project with the Archivio di Stato in Venice (http://vtm.epfl.ch), and numerous other research projects with several national and international institutions. EPFL is committed to build research competence in Digital Humanities; it has recently created an Institute of Digital Humanities and is recruiting world-class talent in this domain. EPFL will also offer a Master of Science in Digital Humanities starting with the academic year 2016/2017.

EPFL, with its main campus located in Lausanne, Switzerland, is a dynamically growing and well-funded institution fostering excellence and diversity. EPFL offers internationally competitive salaries, significant start-up resources, and outstanding research infrastructure.

The EPFL environment is multi-lingual and multi-cultural, with English often serving as a common interface.

Applications should include a cover letter with a statement of motivation, curriculum vitae including a list of publications and patents, concise statements of research and teaching interests, and the names and addresses of at least three references for junior and six for senior positions. Applications must be uploaded in PDF format to the recruitment web site:

https://academicjobsonline.org/ajo/jobs/5387

Formal evaluation of candidates will begin on **June 1, 2015** and will continue until the positions are filled.

Enquiries may be addressed to:

Prof. Thomas David
Search Committee Chair
Email: dhg-search@epfl.ch

For additional information on EPFL, please consult the web sites www.epfl.ch and www.cdh.epfl.ch

EPFL is committed to increasing the diversity of its faculty, and strongly encourages women to apply.

[CONTINUED FROM P. 104] **first formulated the relational model for database management.**

It was pretty obvious from that paper that building a relational system was a good idea, so Gene and I started doing it. Neither of us had ever written a significant software system before, but we persevered and made it work. Ingres was essentially the only relational database system where you could put your hands on the code, and it was very widely run in the late 1970s.

System R, on the other hand—the other working relational database of the era—was proprietary to IBM. What was your relationship with the System R team?

We didn't really work together, but they were 50 miles away and we would get together periodically to exchange ideas. I guess you could say both groups were disciples of Ted Codd. There was no prior art, and I think we all believed we were pathfinders on a mission. Between the two systems, most of the techniques used in relational databases were invented by one project or the other.

One of the techniques you introduced with Ingres is query rewriting—supporting views, protection, and integrity constraints with a unified mechanism.

It was clear to us that you had to support the notion of relational views. There's how the data is stored, and if the user wants to see it differently, then you ought to present him with what's called a virtual relation—a table that doesn't really exist, but can be derived from the actual stored data. I wrote a paper in 1976 that showed how to do it in a very simple way. You just take the user's query and you modify it, and then it runs on stored data, even though you're running a query on data that's not stored. It's high-performance, doesn't take a lot of code, and works.

In 1980, you formed the Ingres Corporation to commercialize your work.

It was clear that if we wanted to make a difference, we had to form a commercial company. If you have innovative technology, the best way to have it see the light of day is to form a start-up.

A few years later, you began work on Ingres's successor, Postgres. Postgres introduced a range of other innovations like the object-relational model, which enables users to extend the database by defining and manipulating their own objects.

In 1983 or 1984, we pushed the original academic Ingres code line off a cliff and started over. That was the genesis of Postgres.

At the time, I was also working with the commercial Ingres guys, who had just implemented the ANSI SQL standard for adding date and time as a data type. I got a call from an unhappy customer. Turned out, the standard said to implement the Julian calendar, so that if you have two dates, and you subtract them, then the answer is Julian calendar subtraction. But he was computing interest on financial bonds, and he ran on a calendar where every month has the same length, because you get the same interest on a bond regardless of how long the month is. In modern times, there are lots and lots of different semantics for time, so if you implement one of them, everyone on a different definition is out of luck. When FedEx does two-day delivery, Sunday doesn't count, and when Wall Street says something settles in five days, they mean trading days, not calendar days. So what this guy wanted was a new data type called bond time. And he was happy to store date and time exactly the way it's stored in the Julian calendar; he just wanted his own definition of subtraction. Postgres allowed him to do that, and it made a huge difference in performance and maintainability.

In 2001, you became an adjunct professor at MIT, where you've been incredibly productive.

When we moved here, MIT had nothing going on in database management whatsoever—no professors, courses, or students. But the nice thing about the Boston area is there are so many universities here, and every single one of them has a database person or two. So an extended research group began to form, with people at Brandeis, Brown, Worcester Polytech, and UMass Boston. The Brown guys wanted to build a stream-processing engine, so we built the prototype and it got commercialized. Then it occurred to me, in the data warehouse market, that column stores were wildly better than row stores, so we built a thing called C-store, which turned into Vertica.

You are currently CTO at three startups and co-director of MIT's Intel Science and Technology Center for Big Data. How do you organize your time?

I fight whatever fire is most pressing. Typically I'm at MIT three days a week, and at one or another startup two days a week, but it varies. There are travel requests and customer visits, and they come on their own schedules. I'm so privileged to be able to talk to customers; they are a great source of problems. They'll tell you why your stuff doesn't work, and then you get to fix it.

Let's talk about the legacy relational database market. You've said—and your career certainly reflects—that "one size fits none" when it comes to database applications.

The database market is in a watershed transition right now. In the 1990s, traditional relational database systems were the answer in search of a question. Right now, the legacy implementations are not good for anything. They will be replaced by new implementations that are largely the result of better ideas like column stores, on the one hand, and changing technology on the other. Very large main memories are now routine, and you can leverage that. There are also all kinds of new problem areas.

I think what's going to happen is that there will be at least a half-dozen database architectures that will be vertical-market specific. It's great to be a practicing professional in an industry in transition. The elephants, as I affectionately call them, will only change when they're threatened. C

Leah Hoffmann is a technology writer based in Piermont, NY.

© 2015 ACM 0001-0782/15/06 $15.00

Watch Stonebraker discuss his work in this exclusive *Communications* video.

DOI:10.1145/2754958

Leah Hoffmann

Q&A
The Path to Clean Data

Michael Stonebraker on Ingres, Postgres, dividing his time between academia and start-ups, and why "one size fits none."

A SERIAL ENTREPRENEUR long before the term became commonplace—and a data geek before the age of big data—ACM A.M. Turing Award recipient Michael Stonebraker pioneered techniques that were not just crucial to making relational databases a reality, but that continue to be used in almost all modern systems. Stonebraker spent the first 30 years of his career at the University of California at Berkeley, where he helped develop the still-popular Ingres relational database management system (DBMS) and the object-relational DBMS PostgreSQL (or Postgres). Since 2001, he has served as an adjunct professor at the Massachusetts Institute of Technology (MIT), while forming a total of six start-ups in 14 years around innovations like stream processing and column stores. Here, he looks back at a few career highlights and forward to the future of database systems.



PHOTO BY JARED LEEDS PHOTOGRAPHY

You graduated from Princeton in 1965 and have said that you only went to graduate school to avoid being sent to Vietnam.

Have you heard of the TV show "Route 66," about two guys who drive around the country and have interesting experiences finding themselves? That's what I wanted to do when I graduated college, but that was not an option because of the Selective Service System. So I went to the University of Michigan to sit out the war.

From there, you went to Berkeley and began your groundbreaking work on database management systems.

I was hired by Berkeley with the idea that I was going to do urban engineering—using analytic techniques to tell people where to put fire stations and so on. But it became clear that getting clean data was agony. I had a mentor, Eugene Wong, who suggested that we look at database management, so we read Ted Codd's pioneering paper.

You refer to the 1970 *Communications* paper that [CONTINUED ON P. 103]

Computing Reviews

Connect with our Community of Reviewers

"The thing I like best is that there are reviewers with wildly different styles and perspectives. This makes CR a rich source of valuable opinions."

- John Artz

Association for Computing Machinery

ThinkLoud

www.computingreviews.com

Discover Your Story

Where every discipline in the computer graphics and interactive techniques community weaves together a story of personal growth and powerful creation. Discover unique SIGGRAPH stories using the Aurasma app, then come to SIGGRAPH and build your own.



[DOWNLOAD AURASMA FROM YOUR APP STORE AND SCAN IMAGE ABOVE]

SIGGRAPH2015
Xroads of **Discovery**

9-13 August 2015

Los Angeles Convention Center

Sponsored by ACM**SIGGRAPH**

The 42nd International Conference and Exhibition on Computer Graphics and Interactive Techniques

s2015.siggraph.org

AURASMA

Google play

App Store