

COMMUNICATIONS

CACM.ACM.ORG

OF THE

ACM

08/2015 VOL.58 NO.08

Programming the Quantum Future

Network Science,
Web Science,
and Internet Science

The Moral Challenges
of Driverless Cars

Permissionless Innovation

Association for
Computing Machinery





■ 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 to secure your preferred location!

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



Are you looking for your next IT job?

Do you need Career Advice?

The **ACM Career & Job Center** offers ACM members a host of career-enhancing benefits:

- A **highly targeted focus** on job opportunities in the computing industry
- **Access to hundreds** of industry job postings
- Resume posting **keeping you connected** to the employment market while letting you maintain full control over your confidential information
- **Job Alert system** that notifies you of new opportunities matching your criteria
- **Career coaching** and guidance available from trained experts dedicated to your success
- **Free access** to a content library of the best career articles compiled from hundreds of sources, and much more!



Visit **ACM's Career & Job Center** at:
<http://jobs.acm.org>



Association for
Computing Machinery

Advancing Computing as a Science & Profession

The **ACM Career & Job Center** is the perfect place to begin searching for your next employment opportunity!

Visit today at **<http://jobs.acm.org>**

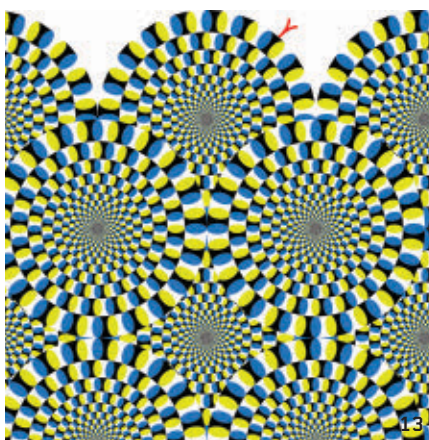
Departments

- 5 **Editor's Letter**
Why Doesn't ACM Have a SIG for Theoretical Computer Science?
By Moshe Y. Vardi
-
- 7 **Cerf's Up**
Invention
By Vinton G. Cerf
-
- 8 **Letters to the Editor**
Not So Easy to Forget
-
- 10 **BLOG@CACM**
Plain Talk on Computing Education
 Mark Guzdial considers how the variety of learning outcomes and definitions impacts the teaching of computer science.
-
- 25 **Calendar**
-
- 95 **Careers**

Last Byte

- 96 **Upstart Puzzles**
Brighten Up
By Dennis Shasha

News



- 13 **Teaching Computers with Illusions**
 Exploring the ways human vision can be fooled is helping developers of machine vision.
By Esther Shein
-
- 16 **Touching the Virtual**
 Feeling the way across new frontiers at the interface of people and machines.
By Logan Kugler
-
- 19 **The Moral Challenges of Driverless Cars**
 Autonomous vehicles will need to decide on a course of action when presented with multiple less-than-ideal outcomes.
By Keith Kirkpatrick

Viewpoints

- 21 **Privacy and Security**
Security for Mobile and Cloud Frontiers in Healthcare
 Designers and developers of healthcare information technologies must address preexisting security vulnerabilities and undiagnosed future threats.
By David Kotz, Kevin Fu, Carl Gunter, and Avi Rubin
-
- 24 **Economic and Business Dimensions**
Permissionless Innovation
 Seeking a better approach to pharmaceutical research and development.
By Henry Chesbrough and Marshall Van Alstyne
-
- 27 **Code Vicious**
Hickory Dickory Doc
 On null encryption and automated documentation.
By George V. Neville-Neil
-
- 29 **Education**
Understanding the U.S. Domestic Computer Science Ph.D. Pipeline
 Two studies provide insights into how to increase the number of domestic doctoral students in U.S. computer science programs.
By Susanne Hambrusch, Ran Libeskind-Hadas, and Eric Aaron
-
- 33 **Viewpoint**
Learning Through Computational Creativity
 Improving learning and achievement in introductory computer science by incorporating creative thinking into the curriculum.
By Leen-Kiat Soh, Duane F. Shell, Elizabeth Ingraham, Stephen Ramsay, and Brian Moore

Practice



44

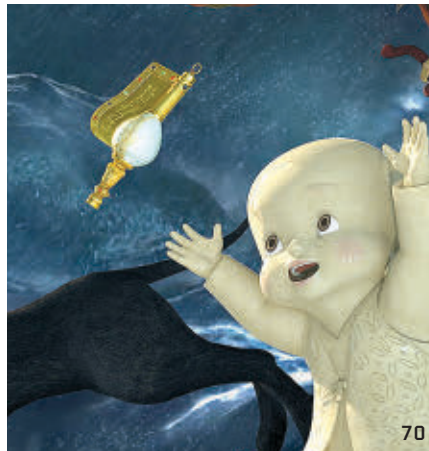
- 36 **Testing Web Applications with State Objects**
Use states to drive your tests.
By Arie van Deursen

- 44 **From the EDVAC to WEBVACS**
Cloud computing for computer scientists.
By Daniel C. Wang



Articles' development led by queue.acm.org

Contributed Articles



70

- 52 **Programming the Quantum Future**
The Quipper language offers a unified general-purpose programming framework for quantum computation.
By Benoît Valiron, Neil J. Ross, Peter Selinger, D. Scott Alexander, and Jonathan M. Smith



Watch the authors discuss their work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/programming-the-quantum-future>

- 62 **Surveillance and Falsification Implications for Open Source Intelligence Investigations**
Legitimacy of surveillance is crucial to safeguarding validity of OSINT data as a tool for law-enforcement agencies.
By Petra Saskia Bayerl and Babak Akhgar

- 70 **Challenges Deploying Complex Technologies in a Traditional Organization**
The National Palace Museum in Taiwan had to partner with experienced cloud providers to deliver television-quality exhibits.
By Rua-Huan Tsai, David C. Yen, and Yu-Chien Chang

Review Articles



76

- 76 **Network Science, Web Science, and Internet Science**
Exploring three interdisciplinary areas and the extent to which they overlap. Are they all part of the same larger domain?
By Thanassis Tiropanis, Wendy Hall, Jon Crowcroft, Noshir Contractor, and Leandros Tassioulas

Research Highlights

- 84 **Technical Perspective**
Corralling Crowd Power
By Aniket (Niki) Kittur
- 85 **Soylent: A Word Processor with a Crowd Inside**
By Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich



Watch the authors discuss their work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/soylent-a-word-processor-with-a-crowd-inside>



About the Cover:
Schrödinger's cat, an iconic image used for decades to illustrate the differences in emerging theories in quantum mechanics, takes a 21st-century spin in this month's cover story (p. 52), where quantum programming languages are explored and a model of quantum computation is presented. Cover illustration by FutureDeluxe.



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 Cappel

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

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

Jianguang 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 Özsu;

Mary Lou Soffa

ACM U.S. Public Policy Office

Renee Doplick, 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

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.

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 acmm mediasales@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 Mehlforn;

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; Terry Coatta;

Stuart Feldman; Benjamin Fried; Pat

Hanrahan; Tom Limoncelli;

Kate Matsudaira; Marshall Kirk McKusick;

George Neville-Neil; Theo Schlossnagle;

Jim Waldo

The Practice section of the CACM

Editorial Board also serves as

the Editorial Board of *ACM Queue*.

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 Bestavros 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:10.1145/2791388

Moshe Y. Vardi

Why Doesn't ACM Have a SIG for Theoretical Computer Science?

Wikipedia defines Theoretical Computer Science (TCS) as the “division or subset of general computer science and mathematics that focuses on more abstract or mathematical

aspects of computing.” This description of TCS seems to be rather straightforward, and it is not clear why there should be geographical variations in its interpretation. Yet in 1992, when Yuri Gurevich had the opportunity to spend a few months visiting a number of European centers, he wrote in his report, titled “Logic Activities in Europe,” that “It is amazing, however, how different computer science is, especially theoretical computer science, in Europe and the U.S.” (Gurevich was preceded by E.W. Dijkstra, who wrote an EWD Note 611 “On the fact that the Atlantic Ocean has two sides.”)

This difference between TCS in the U.S. (more generally, North America) and Europe is often described by insiders as “Volume A” vs. “Volume B,” referring to the *Handbook of Theoretical Computer Science*, published in 1990, with Jan van Leeuwen as editor. The handbook consisted of Volume A, focusing on algorithms and complexity, and Volume B, focusing on formal models and semantics. In other words, Volume A is the theory of algorithms, while Volume B is the theory of software. North American TCS tends to be quite heavily Volume A, while European TCS tends to encompass both Volume A and Volume B. Gurevich's report was focused on activities of the Volume-B type, which is sometimes referred to as “Eurotheory.”

Gurevich expressed his astonishment to discover the stark difference between TCS across the two sides of the

Atlantic, writing “The modern world is quickly growing into a global village.” And yet the TCS gap between the U.S. and Europe is quite sharp. To see it, one only has to compare the programs of the two North American premier TCS conferences—IEEE Symposium on Foundations of Computer Science (FOCS) and ACM Symposium on Theory of Computing (STOC)—with that of Europe's premier TCS conference, Automata, Languages, and Programming (ICALP). In spite of its somewhat anachronistic name, ICALP today has three tracks with quite a broad coverage.

How did such a sharp division arise between TCS in North America and Europe? This division did not exist prior to the 1980s. In fact, the tables of contents of the proceedings of FOCS and STOC from the 1970s reveal a surprisingly (from today's perspective) high level of Volume-B content. In the 1980s, the level of TCS activities in North America grew beyond the capacity of two annual single-track three-day conferences, which led to the launching of what was known then as “satellite conferences.” Shedding the “satellite” topics allowed FOCS and STOC to specialize and develop a narrower focus on TCS. But this narrower focus in turn has influenced what is considered TCS in North America.

It is astonishing to realize the term “Eurotheory” is used somewhat derogatorily, implying a narrow and esoteric focus for European TCS. From my perch as Editor-in-Chief for *Communications*, today's spectrum of TCS is

vastly broader than what is revealed in the programs of FOCS and STOC. The issue is no longer Volume A vs. Volume B or Northern America vs. Europe (or other emerging centers of TCS around the world), but rather the broadening gap between the narrow focus of FOCS and STOC and the broadening scope of TCS. It is symptomatic indeed that unlike the European Association for Theoretical Computer Science, ACM has no Special Interest Group (SIG) for TCS. ACM does have SIGACT, a Special Interest Group for Algorithms and Computation Theory, but its narrow focus is already revealed in its name. In 2014 ACM established SIGLOG, “dedicated to the advancement of logic and computation, and formal methods in computer science, broadly defined,” effectively formalizing the division of TCS inside ACM.

This discussion is not of sociological interest only. The North American TCS community has been discussing over the past few years possible changes to the current way of running its two conferences, considering folding FOCS and STOC into a single annual conference of longer duration. A May 2015 blog entry by Boaz Barak is titled “Turning STOC 2017 into a ‘Theory Festival.’”

I like very much the proposed directions for FOCS/STOC, but I would also like to see the North American TCS community show a deeper level of reflectiveness on the narrowing of their research agenda, starting with the question posed in the title of this editorial: Why doesn't ACM have a SIG for Theoretical Computer Science?

Follow me on Facebook, Google+, and Twitter.

Moshe Y. Vardi, EDITOR-IN-CHIEF

Copyright held by author.

6th Annual ACM SIGPLAN Conference on

Systems,
Programming,
Languages, and
Applications:
Software for
Humanity

Location

Sheraton Pittsburgh at
Station Square Hotel

Events

- 30th Annual OOPSLA
- Onward!
- Wavefront
- Generative Programming: Concepts & Experiences
- Dynamic Languages Symposium (DLS)
- Software Language Engineering (SLE)
- Pattern Languages of Programming (PLoP)
- and more

General Chair

Jonathan Aldrich
Carnegie Mellon University

OOPSLA Papers Chair

Patrick Eugster
Purdue University

Onward! Papers Chair

Gail Murphy
University of British Columbia

Onward! Essays Chair

Guy Steele
Oracle Labs

<http://splashcon.org>
info@splashcon.org





DOI:10.1145/2798333

Vinton G. Cerf

Invention

I am on a brief holiday in the U.K. visiting stately homes and manor houses in Cotswolds. If you have never had an opportunity to visit some these incredibly large and elaborate

dwelling, you might give it some consideration. What I have found most interesting is the mechanical ingenuity of our 16th- and 17th-century ancestors. At one such mansion, I encountered a clever spit-turner before a huge fireplace driven by gravity. One pulled a weight on a rope up to the ceiling and as the weight dropped, it turned a spit. The tricky bit was to control the rate of descent as to turn the spit slowly. A rather clever gearing arrangement used a small gear to turn a larger one to achieve the desired effect.^a Gears were well known by then and used with water wheels and, of course, with clocks.

At the Folger Shakespeare Library in Washington, D.C., there is an exhibit of clocks made by John Harrison^b as well as by others. Harrison is the famous hero of Dava Sobel's *Longitude*,^c the story of the invention and refinement of the ship's chronometer. Harrison was in competition with the so-called Lunar Distance method of estimating longitude. Edmund Halley was the Astronomer Royal at that time and strongly encouraged Harrison's work. After completing three sea clocks (H1-H3), Harrison concluded that a compact watch could perform as well or better. He designed his first sea

watch (designated H4 now) and a voyage was undertaken in 1761 aboard the HMS Deptford from Portsmouth to Jamaica and back. The watch performed well but the Board of Longitude complained this might have been sheer luck and refused Harrison the £20,000 prize that had been offered in the 1714 Longitude Act. A second voyage was undertaken to Bridgeport, Barbados with Harrison's son, William, onboard with H4. Also along on this voyage was the astronomer Reverend Nevil Maskelyne, who carried out the calculations needed for the Lunar Distance method. Both methods worked fairly well but Maskelyne became the Royal Astronomer on return from Barbados and sat on the Longitude Board where he made a very negative report on the performance of the watch. Maskelyne was plainly biased and eventually Harrison turned to King George III for assistance. You must read Sobel's book or watch the Granada dramatization to learn the rest of the story!

There is so much history and drama hidden in some of the mechanical designs in these ancient buildings. The Protestant Reformation began in 1517 with the publication by Martin Luther of his 95 theses. By the time of Henry VIII, England was still Catholic but owing to the refusal of the Pope to annul his marriage to Catherine of Aragon, Henry persuaded the Parliament to pass the Act of Supremacy in late 1534 declaring Henry the supreme head of the Anglican Church. In 1540, the Catholic Church created the Jesuit Order to battle the protestant movement. Jesuit

priests would be spirited across the English Channel to be housed in stately homes of rich Catholic families. By the time of Elizabeth I, it was illegal to practice Catholicism in England and search parties looking for Catholic priests were a regular feature of the time.

Many of the Catholic families had *hides* built into their homes, spaces to hide priests and others simply to hide valuables. At Harvington Hall, the manor house had many such hiding places.^d Some were floorboards that could be tilted to reveal spaces, sometimes stair steps would lift up and in some cases, wall timbers were actually mounted on axles to rotate if you knew where to push. The mechanical inventiveness of these hiding places was notable.

From the Harvington Hall article:

In the late 16th century, when the home became part of a loose network of houses dedicated to hiding Catholic priests, Jesuit builder Nicholas Owen was sent to the building to install a number of secret spots where they could be concealed, should the Queen's men come calling.

Owen built little cubbies hidden behind false attic walls that could be accessed through a fake chimney; a beam that could flip up on an access point revealing a chamber in the walls (which was only discovered 300 years later by some children who were playing in the house); and, most elaborately, a secret room hidden behind another hidden compartment under a false stair. Smaller compartments to hide the priests' tools were also built into the floors.

One could go on for many volumes about the rich state of invention in the past. In our computing world, invention is still the coin of the realm, made all the easier by evolving computing and networking platforms of the present. ■

d <http://slate.me/1Jk1ISL>

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

Copyright held by author.

a I appreciated this design especially because as a young boy in the 1950s, I was designated to turn such a spit by hand in a large brick barbecue behind my home.

b https://en.wikipedia.org/wiki/John_Harrison

c Sobel, D. *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*. Penguin, New York, 1995; ISBN 0-14-025879-5.

Not So Easy to Forget

MEG LETA JONES's Viewpoint "Forgetting Made (Too) Easy" (June 2015) raised an important concern about whether the Court of Justice of the European Union's Google Spain judgment created an extra burden for data controllers like Google and other search engines, though not clear is whether it is being borne out or outweighs the privacy gains for hundreds of millions of users. She wrote Google "...is without any guidance as to which interests should trump others, when, and why." This is not quite true. A number of guiding principles have been published, including from the Article 29 Working Party (the independent advisory body of representatives of the European Data Protection Authorities that would arbitrate disputes under data-protection law) and from Google's own Advisory Council. The European Union's Data Protection Directive also includes a number of defenses against and exemptions from data-protection complaints. There is no reason to believe a clear set of principles will not emerge, especially as Google remains in close touch with Data Protection Authorities, even if more complex cases demand close and exhaustive inspection.

Google is meanwhile developing its own jurisprudence; for example, along with 79 other Internet scholars, I helped write an open letter to Google in May 2015 (<http://www.theguardian.com/technology/2015/may/14/dear-google-open-letter-from-80-academics-on-right-to-be-forgotten>) asking for more transparency, precisely to increase the public's understanding of how the process is administered, so researchers and other data controllers can learn from Google's experience.

Moreover, there is no evidence of a flood of frivolous "de-indexing" requests. Individuals do not "enforce" their right directly with the data controller; rather, they submit requests that can be turned down, and are. Google has fairly consistently rejected about 60% of such requests, with few taken further; for example, in the U.K.,

out of some 21,000 rejected requests for de-indexing as of June 2015, only about 250 have been taken to the next step and referred to the U.K. Information Commissioner's Office.

Also note the right to be de-indexed is not new but a right E.U. citizens have had since the Data Protection Directive was adopted by the European Union in 1995. Surely the pursuit of this right should not have to wait for jurisprudence to develop, especially as the jurisprudence will emerge only if people pursue the right.

Kieron O'Hara, Southampton, U.K.

Author's Response:

The guidelines Article 29 Working Party produced six months after the Court of Justice of the European Union decision (while welcome) are still incredibly vague, point out how varied are the numerous criteria EU member states must follow, and raise additional sources of conflict that deserve more debate and public participation. As for terms like "Google jurisprudence," Google should have no jurisprudence. New rights in light of new technology must be shaped carefully in an international context, evolving through an open, democratic process instead of the dark corners of a detached intermediary.

Meg Leta Jones, Washington, D.C.

Who's Digital Life Is It Anyway?

Serge Abiteboul et al.'s Viewpoint "Managing Your Digital Life" (May 2015) proposed an architecture for preserving privacy while consolidating the data social media users and online shoppers scatter across multiple sites. However appealing one finds this vision, which is similar to one I aired in an *O'Reilly Radar* article Dec. 20, 2010 (<http://oreil.ly/eX2ztY>), a deeper look at the ideal of personal data ownership turns up complications that must be addressed before software engineers and users alike can hope to enjoy implementation of such a personal information management system. These complications involve fairly

well-known questions about who owns the data, particularly when it is shared in the interactions that characterize the modern Internet. Sales data and comments posted to friends' social media sites constitute examples of such data where ownership is unclear, and even a photograph one takes can be considered personal data of the people in the photo. Moreover, while Abiteboul et al. mentioned the benefits of combining and running analyses on data in one's personal repository, they did not address the more common task of how to combine and analyze data from millions of users of a service. Segregated data in repositories maintained by its individual owners would protect those owners from the privacy violations of bulk analyses but also introduce serious hurdles for researchers looking to perform them.

Andy Oram, Arlington, MA

Help Kill a Dangerous Global Technology Ban

Earlier this year, the Electronic Frontier Foundation launched "Apollo 1201," a project to reform Section 1201 of the 1998 Digital Millennium Copyright Act, which threatens researchers and developers with titanic fines (even prison sentences) for "circumventing" access restrictions (even when the access itself is completely lawful) that stifle research, innovation, and repair. Worse, digital rights management, or DRM, vendors claim publishing bug reports for their products breaks the law.

EFF has vowed to fix this.

Law must not stand in the way of adding legitimate functionality to computers. No technologist should face legal jeopardy for warning users about vulnerabilities, especially with technology omnipresent and so intimately bound up in our lives. People who understand should demand an Internet of trustworthy things, not an Internet of vuln-riddled things pre-pwned for criminals and spies.

Though the DMCA has been on the books since 1998, 1201 has hardly been

litigated, giving courts few opportunities to establish precedents and provide clarity to computer scientists, engineers, and security researchers.

1201 advocates—mainly giant entertainment companies—pursue claims only against weak defendants. When strong defendants push back, the other side runs, as when a team led by Ed Felten (then of Princeton, now Deputy U.S. Chief Technology Officer) wrote a paper on a music-industry DRM called the Secure Digital Music Initiative (SDMI). The RIAA threatened Felten and USENIX, at whose August 2001 Security Symposium the paper was to be presented.

The Electronic Frontier Foundation took Felten's case, and the RIAA dropped the threat and disavowed any intention to pursue Felten over SDMI. It knew the courts would reject the idea that record executives get a veto over which technical articles journals are able to publish and conferences can feature.

It is time to bring 1201's flaws to court. EFF is good at it. One of its seminal cases, *Bernstein v. United States*, struck down the NSA's ban on civilian access to crypto, arguing the code is a form of expressive speech entitled to First Amendment protection. EFF looks forward to proving that banning code still violates the First Amendment.

That is where ACM members come in. EFF is seeking academic researchers and professors whose work is likely to attract threats due to 1201. If someone in your lab or department is working on such a project (or gave it up over fear of litigation) EFF is interested in hearing about it.

The legitimacy and perceived efficacy of 1201 is an attractive nuisance, inviting others to call for 1201-like protections for their pet projects.

FBI Director James Comey has called for backdoors on devices with encrypted file systems and communications. As ACM members doubtless understand, there is no way to sustain a backdoor without some legal prohibition on helping people install backdoor-resistant code.

EFF is not just litigating against 1201; working with a global network of organizations, EFF is able to lobby the world's governments to rescind their own versions of 1201, laws passed at the insistence of, say, the U.S. Trade Representative.

Time is running out. Please get in touch and help us help you kill 1201.

Cory Doctorow, London, U.K.

What Grounds for Jahromi Release?

Jack Minker's letter to the editor "Bahrain Revokes Masaud Jahromi's Citizenship" (Apr. 2015) cited "attending a rally on behalf of freedom" as an illegitimate reason for the imprisonment of someone he supports. All are in favor of "freedom," of course, and would happily attend rallies seeking such a universal goal. But not all those seeking freedom are laudable. Most prisoners and lawbreakers would like to have freedom. Many terrorists call themselves "freedom fighters." It is not enough to proclaim innocence by saying a person is seeking freedom. It is necessary to be more specific and comprehensive. Perhaps the person on whose behalf Minker advocates does indeed deserve to be free. But Minker's description of the problem was insufficient to convince one that is the case.

Robert L. Glass, Brisbane, Australia

Author's Response

Glass assumes peaceful protesters (and Jahromi perhaps imprisoned following a trial with due process), as might be expected in Australia. This is not the situation in Bahrain. For a description of the repressive and atrocious human rights situation in Bahrain, see U.S. Department of State Universal Periodic Reviews 2011–2015 (<http://www.state.gov/j/drl/upr/2015>) and the report of the Bahrain Independent Commission of Inquiry (<http://www.bici.org.bh>).

Jack Minker, College Park, MD

Validity Seal of Approval for Every Program

Lawrence C. Paulson's letter to the editor "Abolish Software Warranty Disclaimers" (May 2015) on Carl Landwehr's Viewpoint "We Need a Building Code for Building Code" (Feb. 2015) addressed only a minor factor in user-experienced angst. Any individual program includes few bugs on its own. But when a user invokes a suite of programs, it is the logic, arithmetic, and semantic incompatibilities among the programs that result in system-level er-

rors and abortions. The purveyor of any of these programs cannot guarantee the progress and safety properties of the subsequent user-formed system will be valid. Software developers and users alike need the equivalent of the Good Housekeeping Seal of Approval for each vendor program, as well as a way for users to assess the risks they create for themselves when choosing to make programs interoperate. Moreover, users must be able to do this each and every time thereafter when anyone performs "maintenance" on a program or dataset in the user-specific ensemble.

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 0001-0782/15/08 \$15.00

Coming Next Month in COMMUNICATIONS

Commonsense Reasoning and Commonsense Knowledge in Artificial Intelligence

Bitcoin: Under the Hood

Experiments as Research Validation: Have We Gone Too Far?

Theory Without Experiments

Trustworthy Hardware from Untrusted Components

Language Translation at the Intersection of AI and HCI

Plus the latest news about sensing emotions, the leap second, and new aggregators for mobile users.

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.



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

DOI:10.1145/2788449

<http://cacm.acm.org/blogs/blog-cacm>

Plain Talk on Computing Education

Mark Guzdial considers how the variety of learning outcomes and definitions impacts the teaching of computer science.



Mark Guzdial
"The Babble of Computing Education: Diverse Perspectives, Confusing Definitions"
<http://bit.ly/1LFwTbT>

May 22, 2015

Recruiting season for new faculty is drawing to a close, and I felt a twinge of jealousy for non-education fields. A robotics researcher can claim, "My robot is able to do this task faster and more reliable than any other," and a general computer science (CS) faculty audience can agree that task is worth doing and doing better. An HCI researcher can say, "People can achieve this goal better and faster with my system," and the general CS faculty audience can see why that is important. When a computing education researcher says, "I now have new insights into how people understand X," the general CS faculty audience often does not know how to measure the value of that insight. "But can the students *program* using X?" and "Can they *design* systems using X?" and "X is only important in domain Y, not in the domain Z. How do you know if they can work in domain Y?"

Part of the problem is that general CS faculty do not understand education research. In social sciences, developing an insight, a hypothesis, or a "pre-theoretical" exploration is often a *lot* of work, and it is a contribution even before it is a theory, a model, or an intervention to improve some desired educational outcome.

A bigger problem is that we have many different learning outcomes and definitions in computing education. I recently told a colleague at another institution about our BS in Computational Media at the Georgia Institute of Technology, which may be the most gender-balanced ABET-accredited computing program in the U.S. (see discussion of enrollment at <http://bit.ly/1IdgDR2>, and of graduation rates at <http://bit.ly/1eNHsyE>). The response was, "That's great that you have more women in computational media, but I want them in CS."

The U.S. National Science Foundation (NSF) has been promoting two computer science courses in high schools across the country: Exploring CS (ECS, see <http://www.exploringcs.org/>) and the new Advanced Placement in Computer Science Principles (AP

CSP, see <http://apcsprinciples.org/>). ECS and AP CSP are getting significant adoption across the U.S. Because high school education is controlled by the individual states (which may delegate control down to the individual districts), not the federal government, it is difficult to make sure it is the same course in all those states. I am part of an NSF Alliance that works with states to improve their computing education (Expanding Computing Education Pathways Alliance, ECEP, <http://ecepalliance.org/>), and I am seeing states struggling with wanting to adopt ECS and AP CSP, but also wanting to fit the courses to their values and needs.

In one state we work with in ECEP, the state decided to create its own version of Exploring CS. They kept all the same technical material, but they de-emphasized inquiry learning and equitable access to computing, in order to increase the technical content. They also wanted to ensure that all high school students learn about database systems, because that was important for IT jobs in the state (they are now reconsidering this decision).

In a couple of states, there will be CS Principles and AP CS Principles as

two different courses. One reason for the difference will be the performance tasks in the AP CSP. The definition for the new Advanced Placement exam in CS Principles will include evaluation of activities in the classroom, where students describe code, create products, and demonstrate that CS is a creative and collaborative activity. It is a great idea, but implementing it takes time. The performance tasks take about 23 hours of classroom time to complete. With practice for the tasks (you would not want your students to be evaluated on the first time they tried something), AP CSP may cost a class 50 hours or more of time, which might be spent on more instructional time.

Exploring CS was developed in high school, but in some states is being adopted into middle schools. Middle schools differ even more dramatically than high schools. Some of the efforts to adopt ECS involve integrating it into multidisciplinary science and mathematics courses or units, which is common in the new Next Generation Science Standards implementations (see California examples at <http://bit.ly/1GjdgI1>). A focus on equity in computing is difficult to sustain when computing is just one of the several disciplines integrated in the course. Is the course still ECS then?

We have a babble of conflicting goals, definitions, and learning outcomes around computing education. I regularly hear confusion among the state education policymakers with whom we work in ECEP:

► “Some people tell me that there’s computer science without programming, and other people say programming is key to computer science. Which is it?”

► “We go beyond programming in our CS classes. In our first class, we teach students how companies manage their IT customer service. Isn’t that computer science? Why not?”

► “I want every student in my state to know how to write SQL for Oracle database because that’s an important job skill. Should I build that into our state’s version of ECS or our version of CSP?”

► “The workers in our state need to learn applications and tools. I see that in the job ads. I don’t see any-

“As a computing education researcher, I like the babble. I like many different possibilities being explored at once.”

body advertising for ‘computational thinking.’”

► “Can you teach computer science to our special needs students? To our English-Language Learner students? If you want to teach computer science to *everyone*, you have to cover *everyone*.”

► “Here’s our curriculum. What has to go to make room for computer science?”

Just recently, the organization that owns Advanced Placement, the College Board, struck partnerships with Code.org (<http://usat.ly/1Q4is7b>) and with Project Lead the Way (PLTW, <http://bit.ly/1M265SG>) to endorse their curricula for AP CS Principles. This is a major move to consolidate AP CSP curricula. The whole purpose for doing a new AP CS exam is to make it more accessible, to get more schools to offer it, and to reach schools and students that did not previously have access to CS education. If you were a principal of a school that never offered AP CS before and you have to pick a curriculum, wouldn’t you pick a College Board-endorsed curriculum first? PLTW is expensive for schools to offer, unlike Code.org’s curriculum, which is free. Both offer teacher professional development (PD), but Code.org pays for part of their teacher PD. It is likely Code.org’s curriculum will become the de facto standard for CS Principles.

That may be the right thing to grow computing education. Diverse perspectives are really valuable, but they are also confusing. Most school administrators do not know what CS is. The College Board is not preventing

other CSP curricula. By backing a couple of approaches, it becomes easier to figure out, “What is CS Principles? Ohhh—that’s CS Principles.” A little less babble may go a long way toward increasing access.

As a computing education researcher, I *like* the babble. I like many different possibilities being explored at once. I like a diversity of curricula, and schools with different values implementing the curricula in different ways. As a computing education advocate, I understand the education system can only withstand so much babble—especially at these early stages, when computing is still misunderstood. The teachers, principals, administrators, and policymakers who run these systems need definitions to help them understand computing. It is difficult for computer scientists to agree on these definitions. Maybe the College Board and Code.org will do it for us.

Many thanks to Barbara Ericson, Renee Fall, and Rick Adrion for comments and additions on this blog post. While our work in ECEP is funded by the U.S. NSF, the opinions expressed here are my own and may not represent NSF’s or the rest of ECEP’s.

Comments

Is there a de facto standard curriculum for AP CSA? There does not seem to be one, and that is probably good. AP CSA evolved a lot over the years while AP CSP is brand new. With code.org investing big in the largest districts, I can see the temptation to assume it will become the standard. I suspect though that schools that already have CS, especially CS without AP CSA, may want to create their own to fit their specific school environment. I think (maybe hope is more accurate) that some schools will want to experiment with different tools, languages, and curriculum in their version of AP CSP.

—Alfred Thompson

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

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

Teaching Computers with Illusions

Exploring the ways human vision can be fooled is helping developers of machine vision.

EARLIER THIS YEAR, debate over the color of a dress set the Internet ablaze with discussion over why people were viewing the same exact image, yet seeing it differently. Now throw computers in the mix; unlike humans, who see certain images differently, machines register and recognize visual images on another level altogether. What humans see is determined by biology, vision experts say, while computers determine vision from physical measurements.

While the two fields can inform one another, researchers say more work needs to be done to teach computers how to improve their image recognition.

Those efforts are important because we want machines such as robots to see the world the way we see it. "It's practically beneficial," says Jeff Clune, assistant professor and computer science director of the Evolving Artificial Intelligence Lab at the University of Wyoming. "We want robots to help us. We want to be able to tell it to 'go into the kitchen and grab my scissors and bring them back to me,'" so a robot has to be taught what a kitchen looks like, what scissors are, and how to get there," he says. "It has to be able to see the world and the ob-



Google's winning entry in the 2014 ImageNet competition helps computers distinguish between individual objects.

jects in it. There are enormous benefits once computers are really good at this."

Yet no matter how good machines might get at recognizing images, experts say there are two things they are lacking that could trip them up: experience and evolution.

Computers have already gotten pretty good at facial recognition, for example, but they "will never understand the nuances we grasp right away when we see a face and access all the information related to that face," says Dale Purves, a neurobiology professor at Duke University. People, on the

other hand, “have a ton of information based on what that face means to us ... and we immediately understand the behavioral implications of a frown or a smile.” Getting to all that, he says, will be a long struggle for machine vision “because machines so far don’t know what’s important for behavioral success in the world and what’s not.”

In contrast, humans have grasped those nuances based on millions of years of evolution, as well as individual experience, Purves notes. “Many people have said in many elegant ways that nothing in biology makes sense, except in the light of evolution. I think that’s exactly right. Machine vision fails to recognize that dictum.”

People are trying to get artificial systems to see the world as it is, “whereas for our brain, the way our nervous system evolved through the ages is not necessarily to see the world as it is—it’s to see the world in a way that has made our survival and our reproduction more likely,” adds Susana Martinez-Conde, a professor and director of the Laboratory of Integrative Neuroscience at the State University of New York (SUNY) Downstate Medical Center.

The human brain makes “a lot of guesstimates,” explains Martinez-Conde, whose work focuses on visual perception, illusions, and the neurological basis for them. “We take lim-

ited information from the reality out there and fill in the rest and take shortcuts and arrive at a picture that may not be the perfect match with what’s out there, but it’s good enough.”

One well-known example of an illusion humans and machines register differently is that of rotating snakes (<http://bit.ly/1IRuVDb>). Martinez-Conde says the image is actually stationary, but appears to move when viewed on paper, “because [of] the way our motion sensitivity circuits in the brain are put together or work in such a way that when you have a certain sequence [it] is interpreted as motion, even though there’s no actual motion in the image.”

The human brain has vision neurons that specialize in detecting motion, and that is what the majority of people will see when they view the image, she says. However, age plays a role in what people see as well.

Because the snake illusion is relatively new, what is still not well understood is why people who are about 40 years old or younger are more likely to see motion, but those who are 50 years and older tend not to see it, Martinez-Conde notes. No one knows yet why the motor system experience changes as people age, she says. “The interesting thing is, the motor visual system deteriorates with age, and [yet] you tend to see more reality than illusion. Seeing

motion in the [snake] illusion is a sign your visual system is healthy.”

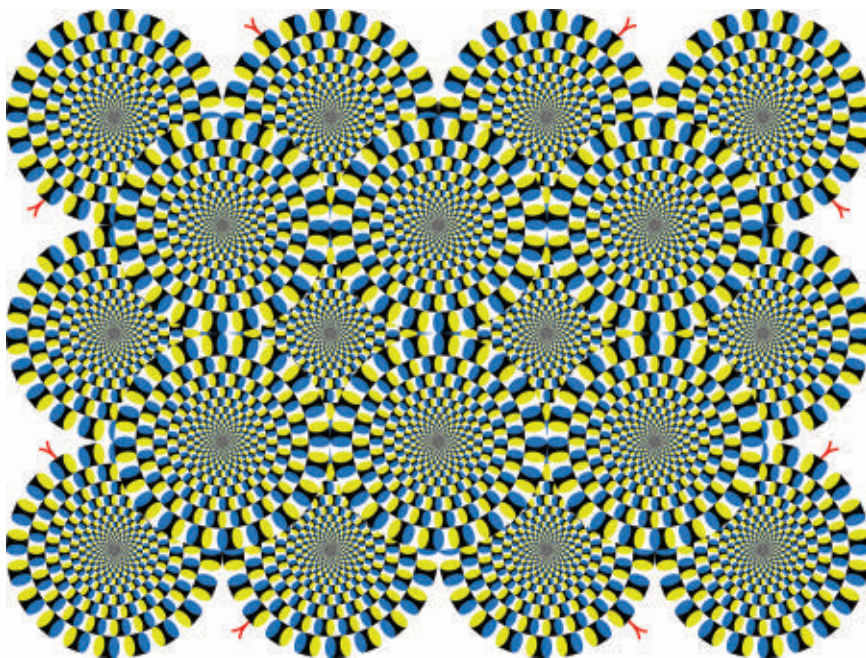
Machine vision, on the other hand, is based on algorithms that can measure items in the environment and use them in driverless cars and elsewhere, says Purves. Humans do not have access to the same information that machine algorithms depend upon for vision.

“We human beings ... have a very deep problem, being that we can’t get at the physical world because we don’t have ways of measuring it with apparatus like laser scanners or radar or spectrophotometers, or other ways to make measurements of what’s physically out there,” he says. Yet, “everyone admits we do better in face recognition and making decisions than machines do, using millions of years of evolutionary information” on a trial-and-error basis.

That does not stop people from trying to get humans and machines closer to seeing the same illusions. Kokichi Sugihara, a mathematician at Meiji University in Tokyo, has been working on a program that will enable computers to perceive depth in 2D drawings. His interest is to “allow a computer, by processing information input, to understand a 3D shape based on a projection drawn with lines, he writes on the university’s website (<http://bit.ly/1CAwz7F>).

“A computer often fails to reconstruct a 3D shape from a projection drawing and delivers error messages, while humans can do this task very easily,” Sugihara writes. “We visualize a 3D object from a 2D drawing based on the preconceived assumption that is obtained through common sense and visual experience ... however, the computer is not influenced by any assumption. The computer examines every possibility in order to reconstruct a 3D object and concludes that it is ‘able to do it.’”

There are different methods that can be used to “fool” computer algorithms so what systems and humans see is more closely aligned. One way to enhance artificial vision is to further study what our brains see, says Martinez-Conde. “We know, after all, they work well enough and our visual system is pretty sophisticated, so having a deeper understanding of our visual sys-



The rotating snakes illusion, as presented by Ritsumeikan University professor Akiyoshi Kitaoka.

tem from a neuroscience perspective can be helpful to improving computer vision.” She adds, however, that our visual system “is by no means perfect, so if we got to a point where computer vision is almost as good, that wouldn’t mean the work is done.”

Humans have used natural selection to incorporate in the neural networks in our brains every conceivable situation in the world with visual input, says Purves. “Once computers do that and evolve, in principal they should be as good as us, but it won’t be in visual measurements; they’re coming at [vision] from a very different way. There’s going to be a limit that will never get them to the level at which human beings operate.”

Yet machines can continue to be improved. “If you want to make a really good machine, evolve it” through trial-and-error experiences and by compiling those experiences in their artificial neural circuitry, says Purves. “There’s no reason that can’t be done; you just have to feed them the information that we used to evolve a visual system.” He estimates that in 20 years’ time, machine vision could be as good as human vision, once vision scientists are able to figure out how to evolve an artificial neural network to “survive” in environments “that are as complicated as the world we live in.”

“Humans and computers see things very differently, and there is a lot more for us to do to figure out how these networks work,” agrees Clune. One troubling issue he addressed in a paper is that if a computer identifies a random, static image as, say, a motorcycle, with 100% certainty, it creates a security loophole, he says. “Any time I could get a computer to believe an image is one thing and it’s something else, there are opportunities to exploit that to someone’s own gain.”

For example, a pornography company may produce images that appear to Google’s image filters like rabbits, but which contain advertisements with nudity; or, a terrorist group could get past artificial intelligence filters searching for text embedded in images by making those images appear to the AI as pictures of flowers, he explains. Biometric security features are also potentially vulnerable; “a terrorist could wear a mask of transparent,

plastic film that has static printed on it that is not visible to humans, but could trick a facial recognition system into seeing an authorized security agent instead of recognizing a known terrorist,” Clune says.

While some believe one system could be fooled by certain images whereas another system trained to recognize them would not be, surprisingly, that is not always the case, he says. “I can produce images with one network and show them to a completely different network, and a surprising number of times the other network is fooled by the same images. So there really are some deep similarities in how these computer networks are seeing the world.”

There is no good way yet to prevent networks from being fooled by nefarious means, Clune says, but when the technology improves, “security holes will be closed and they’ll become smarter and more effective,” he says. “They’ll also do a better job when they encounter substantially different situations than they were trained on.”

Today robots can be trained on one type of image from the natural world, but if they encounter images that are too different, they break down and behave in strange and bizarre ways, Clune says. “They need to be able to see the world and know what they’re looking at.” **C**

Further Reading

Purves, D. and Lotto, R. B.

Why We See What We Do: An Empirical Theory of Vision. 2011 ISBN-10: 0878935967

Macknik, S.L. and Martinez-Conde, S.

Sleights of Mind: What the Neuroscience of Magic Reveals About Our Everyday Deceptions. 2010 Henry Holt and Company, LLC. ISBN: 978-0-8050-9281-3

Sugihara, K. (1986).

Interpretation of Line Drawing. MIT Press, Cambridge. <http://www.evolvingai.org/publications>

Nguyen, A., Yosinski, J., and Clune, J. (2015)

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In Computer Vision and Pattern Recognition (CVPR '15), IEEE, 2015.

Esther Shein is a freelance technology and business writer based in the Boston area.

© 2015 ACM 0001-0782/15/08 \$15.00

ACM Member News

THE EXCITEMENT OF DATABASE SYSTEMS



Timos Sellis, a professor in the School of Computer Science and Information Technology at

RMIT University in Melbourne, Australia, finds database systems exciting. “What attracts me most is the constant evolution to provide solutions to people’s needs to manage data of varying complexity, volume, and relationships.”

Sellis’ research focuses on databases, data management, and big data analytics. He says databases in the 1980s dealt mostly with financial data, before coming to include images, geospatial data, and complex data from mobile devices and sensors, giving rise to new analytic requirements based on new data types and the increasingly diverse environments in which data is produced.

After receiving an undergraduate degree in engineering from the National Technical University of Athens, Sellis earned a master’s degree in computer science from Harvard University, and a Ph.D. in computer science from the University of California, Berkeley.

In 2014, Sellis spearheaded the launch of RMIT’s Data Analytics Lab, aimed at training a new generation of big data analytics researchers. “We’re promoting an environment of networking with other research centers, labs, and industry partners, at the national and international level,” he says.

The lab applies analytics to data from a broad range of industries, to foster the emergence of data value chains. “We’re looking holistically at user and text analytics, which have a huge potential to transform the efficiency and productivity in many areas of the economy,” he explains. “User analytics in Smart Cities can infer a person’s activity based on their spatiotemporal footprint in the city or in common areas such as shopping malls, to offer personalized services.”

—Laura DiDio

Touching the Virtual

Feeling the way across new frontiers at the interface of people and machines.

RESearchers have been hard at work redefining the human-machine interface, in particular looking at new ways we can interact with computers through touch, without actually touching something.

Holograms are not new technology, but there is a futuristic frisson surrounding the topic. A computer-generated hologram is created by a sequence of three-dimensional (3D) images that are processed into a virtual image, a visual illusion. If you try to touch one, your hand will go through it.

What is new is the concept of *touchable* holograms: not just projected into the air, and not just superimposed onto an actual object, but “haptic holograms” that you can not only touch, but interact with and move. Computer haptics are the systems required—both hardware and software—to render the touch and feel of virtual objects. Haptic holograms take this one step further: you can now “touch” a 3D projection, a virtual object, and actually feel it.

Haptic holograms create virtual objects that have a digital interface, an interface that is feel-able as well as visible, by sculpting sound to make visible digital features feel like physical ones. The virtual 3D haptic shape becomes a tactile holographic display.

A Touching Story

The skin covering the hand is packed with receptors that communicate tactile feedback for light touch, heavy touch, pressure, vibration, hot and cold, and pain. This helps the brain understand subtle tactile details: smoothness, hardness, density, weight, and so on.

Ultrasound creates vibrations in the air, projected at a set distance to match the surface of the hologram. The skin “feels” these vibrations at different wavelengths to simulate softness/hardness and more. This information enables a virtual, 3D image to be “touched.”



Ultrasound is focused to create the shape of a virtual sphere that may be felt.

To assist in the design and development of tactile interface applications, Marianna Obrist, a visiting researcher at Newcastle University and Lecturer in Interaction Design at the University of Sussex, and her colleagues created a tactile vocabulary of 14 categories of haptic feedback, such as prickly/tingling, coming/going, and pulsing/flowing. “Pulsing” is a 16Hz vibration that stimulates the Meissner’s corpuscle receptors in the skin that are responsible for sensitivity to light touch.

Sriram Subramanian, professor of Human-Computer Interaction in the Computer Science Department at Bristol University, co-directs the Interaction and Graphics Group. The group, led by research assistant Ben Long, developed the UltraHaptics (<http://ultrahaptics.com/>) system, which creates haptic feedback in mid-

air (https://youtu.be/H6U7hI_zIyU). Waves of ultrasound displace the air, causing a pressure difference. When multiple waves arrive at the same place simultaneously, a noticeable pressure difference is created at that point. According to Long, “Touchable holograms, immersive virtual reality (VR) that you can feel, and complex touchable controls in free space are all possible ways of using this system.”

The addition of a Leap Motion (<https://www.leapmotion.com/>) controller—an infrared sensor that tracks the precise position of a user’s fingers in 3D space—enables ultrasound to be directed accurately at a user’s hands to produce the sensation of touch, creating the impression of exploring the surface of an object, which enhances VR. The group is working on using more complex

shapes with greater detail, possibly by having a greater number of smaller speakers to improve the resolution.

A key challenge, Subramanian explains, is in “understanding in greater detail how users perceive haptic feedback with a mid-air haptic system like ours, which is able to target multiple mechanoreceptors simultaneously. This will play an important role in improving the fidelity of the tactile feedback perceived.”

Subramanian envisions a range of applications for touchless haptics. This includes applications in automotive dashboards in which the user can interact with the dashboard without taking their eyes off the road (for example, the user can wave their hand in front of the dashboard to sense feelable knobs and dials).

Another key application is in VR in developing lightweight, high-resolution head-mounted displays. The Bristol group believes 3D haptics can play a “huge role” in increasing user immersion in VR games and applications. Their spin-off company, Ultrahaptics, is focusing on embedding the technology in a number of different products ranging from alarm clocks to home appliances to cars.

Researchers at the University of Tokyo, led by Hiroyuki Shinoda, created the Airborne Ultrasound Tactile Display which, for example, allows a user to feel hologram raindrops bouncing off their hands. Later, Yasuaki Monnai, project assistant professor of the university’s Department of Creative Informatics, and his colleagues at the Shinoda-Makino Lab created a 2D touchscreen floating in 3D. Hapto-Mime (<http://youtu.be/uARGRlpCWg8>) uses both ultrasound waves and infrared sensors to give hands-free tactile feedback. The interface is a virtual, holographic display on an ultra-thin, floating reflective surface. Ultrasound exerts a mechanical force where the beam is focused, which allows a virtual object, such as a piano keyboard or an ATM number pad, to be felt, as changes in ultrasonic pressure give the illusion of different touch sensations.

The Tokyo team is particularly interested in electromagnetic wave propagation and transmission systems, applying them to wireless com-

The Bristol group believes 3D haptics can play a “huge role” in increasing user immersion in VR games and applications.

munication, measurement, and human-machine interfaces. According to Monnai, they anticipate guiding human motions using the virtual image and force. “In our current system, users touch the hologram, but in future, it is also possible that the hologram touches users. This will enable, for example, having a virtual sport coach who tells you how to move your body by stimulating you with visual and haptic sensations at the correct timing and position.”

The haptic feedback in their system is currently quite weak in strength. To present greater tactile sensation, the Tokyo team has modulated the temporal sequence of the force, as vibration is felt more vividly than stationary force, temporal sequence, or a waveform (burst, continuous wave, or other). Monnai stresses the importance of matching the sensation to the visual image, which is hampered by the lack of well-established guidelines for such design. The team continues to seek a way to generate a stronger force, and Monnai says they intend to extend both visual and tactile images from 2D to 3D, and then they hope to design a 3D touchable hologram that allows for both active and passive interaction.

Michael Page, an assistant professor in Toronto’s OCAD University faculty of art, currently is working on porting medical and scientific data to the new generation of holographic technology, with some of the data being multiplexed so it is interactive (touchable). His team created simulation tools for medical students, providing an auto-stereoscopic visual technology that

projects 3D images without requiring the viewer to wear glasses.

“Holograms are at the very top of auto-stereoscopic volumetric viewing systems. No other medium provides a higher sense of realism,” says Page. One big challenge is creating the “self-contained viewing system for the holograms.”

Sean Gustafson, a Ph.D. researcher (who has since graduated) at the Hasso Plattner Institute in Germany, worked on novel spatial interactive technology, such as placing an imaginary iPhone on the palm of your hand. Patrick Baudisch and colleagues at Potsdam University have continued this work to explore other “imaginary interfaces” (screenless ultra-mobile interfaces), notably looking at tactile, spatial, visual, and sensed cues.

Rather than optical illusions, Baudisch and his team are now “betting on the real thing” by working toward developing personal fabrication equipment that works at interactive rates, ultimately in close to real time. Their WirePrint device, a collaboration between Hasso Plattner Institute and Cornell University, prints 3D objects as wireframe previews for fast prototyping by extruding filaments directly into 3D space instead of printing layer-wise. As Baudisch explains, “users interact by interactively manipulating the 3D shape of the device, while most of the know-how comes from the machine, supporting users in the process.”

Another approach is that of “data gloves.” As Patrik Göthe at Chalmers University in Sweden says, “The next paradigm of consumer technology is mostly about seamlessly moving the screen interface to our surroundings as holograms and projections.” His concept is for a partial glove, covering thumb and index finger, attached to the wrist. It can interact with a holographic keyboard via touch-sensitive fingertips.

Haptic feedback in general is likely to become mainstream. Apple has announced its Force Touch trackpad on the early 2015 MacBook Pro, declared to be “a tour de force of engineering” by *AppleInsider* magazine (<http://bit.ly/1FKi2bg>). You will feel clicks as small taps on your finger. Pressure-sensing APIs can also enable you to write your signature on the trackpad, with great-

er pressure creating broader strokes. Haptic feedback is already integrated into iMovie's 10.0.7 update, and more apps for OS X and other iOS devices are undoubtedly in the pipeline.

Touching the Future

Touchable 3D holograms can extend the use of touch interaction to unconventional situations. Here are some hands-on applications:

Real estate—The latest developments in haptic feedback enable potential purchasers to actually touch the textures in a home when viewing a digital tour: rough stone walls, smooth marble, and so on. At the moment, these textures are only available on a sample-size scale, rather than as part of a life-size set, but the technology is still a useful sales tool.

Medical examinations—When haptic holograms are combined with computerized tomography (CT), magnetic resonance imaging (MRI), and ultrasound scans of an area of the body, surgeons will be able to “feel” a tumor, for example, in advance of a live operation. This technology is already available commercially from companies such as RealView Imaging in Israel, which has developed medical holography for interventional cardiology and diagnostic imaging.

Touchable 3D holograms can extend the use of touch interaction to unconventional situations.

Other potential applications include: museum visitors holding irreplaceable artifacts; sticky-fingered chefs checking recipes; users avoiding the germs invariably found on ATM keypads and other public touch electronic devices, and visually impaired users feeling their virtual interface. There is enormous potential for exploitation of the technology in the military, security, and education sectors, as well as in the arts.

Light Years Ahead

According to the January 2015 Holographic Display Market report by MarketsandMarkets, the touchable display market will experience a compound annual growth rate of more than 30% to reach \$3.57 billion by 2020. Researchers feeling their way toward previously

undreamed-of haptic hologram solutions should feel encouraged. **C**

Further Reading

Gustafson, S. (2013).

Imaginary interfaces. Doctoral dissertation, Hasso Plattner Institute, University of Potsdam.
<http://bit.ly/1IL02DG>

Gustafson, S., Holz, C., and Baudisch, P. (2011). Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of UIST 2011*, 283-292.
<http://bit.ly/1yuQ8Tu>

Hoshi, T., Takahashi, M., Nakatsuma, N., and Shinoda, H. (2009).

Touchable holography. *Proceedings of SIGGRAPH 2009*.
<http://bit.ly/1JKL4ju>

Long, B., Seah, S.A., Carter, T., and Subramanian, S. (2014).

Rendering volumetric haptic shapes in mid-air using ultrasound. *ACM Transactions on Graphics*, vol. 33.
<http://bit.ly/1DHqrlr>

Obrist, M., Seah, S.A., and Subramanian, S. (2014).

Talking about tactile experiences. *Proceedings of ACM CHI 2013 Conference on Human Factors in Computing Systems* (pp. 1659-1668). Paris, France.
<http://bit.ly/1yrnCln>

Logan Kugler is a freelance technology writer based in Tampa, FL. He has written for over 60 major publications.

© 2015 ACM 0001-0782/15/08 \$15.00

Milestones

Computer Science Awards, Appointments

CRA HONORS JAHANIAN, GATES

The Computing Research Association (CRA) has named Farnam Jahanian, vice president for research at Carnegie Mellon University (CMU), recipient of the 2015 CRA Distinguished Service Award, bestowed upon one “who has made an outstanding service contribution to the computing research community” in government affairs, professional societies, publications, or conferences, and whose leadership “that has a major impact on computing research.”

Farnam has served as U.S. National Science Foundation assistant director for Computer & Information Science & Engineering. He also has served as co-chair of

the Networking and Information Technology Research and Development subcommittee of the National Science and Technology Council Committee on Technology.

CRA also named Ann Quiroz Gates, chair of the department of computer science at the University of Texas at El Paso, to receive the 2015 A. Nico Habermann Award, bestowed upon individuals who have made “outstanding contributions aimed at increasing the numbers and/or successes of underrepresented groups in the computing research community.”

For over two decades, Gates has been a leader in initiatives supporting Hispanics and members of other underrepresented groups in

the computing field, including leading the Computing Alliance of Hispanic-Serving Institutions, and participating in organizations including the Society for Advancement of Hispanics/Chicanos and Native Americans in Science, the Center for Minorities and People with Disabilities in IT, and the AccessComputing Alliance.

PAPADIMITRIOU RECEIVES 2015 EATCS AWARD

The European Association for Theoretical Computer Science (EATCS) selected Christos Papadimitriou, C. Lester Hogan Professor of Electrical Engineering and Computer Science in the Computer Science division of the University of

California at Berkeley, to receive the EATCS Award.

The organization said Papadimitriou’s body of work has had a profound and lasting influence on many areas of computer science, contributing “truly seminal work” to fields including algorithmics, complexity theory, computational game theory, database theory, Internet and sensor nets, optimization, and robotics.

Papadimitriou also has written textbooks on the theory of computation, combinatorial optimization, database concurrency control, computational complexity, and algorithms, helping to inspire generations of computer scientists.

The Moral Challenges of Driverless Cars

Autonomous vehicles will need to decide on a course of action when presented with multiple less-than-ideal outcomes.

EVERY TIME A car heads out onto the road, drivers are forced to make moral and ethical decisions that impact not only their safety, but also the safety of others. Does the driver go faster than the speed limit to stay with the flow of traffic? Will the driver take her eyes off the road for a split second to adjust the radio? Might the driver choose to speed up as he approaches a yellow light at an intersection, in order to avoid stopping short when the light turns red?

All of these decisions have both a practical and moral component to them, which is why the issue of allowing driverless cars—which use a combination of sensors and pre-programmed logic to assess and react to various situations—to share the road with other vehicles, pedestrians, and cyclists, has created considerable consternation among technologists and ethicists.

The driverless cars of the future are likely to be able to outperform most humans during routine driving tasks, since they will have greater perceptive abilities, better reaction times, and will not suffer from distractions (from eating or texting, drowsiness, or physical emergencies such as a driver having a heart attack or a stroke).

“So 90% of crashes are caused, at least in part, by human error,” says Bryant Walker Smith, assistant professor in the School of Law and chair of the Emerging Technology Law Committee of the Transportation Research Board of the National Academies. “As dangerous as driving is, the trillions of vehicle miles that we travel every year means that crashes are nonetheless a rare event for most drivers,” Smith notes, listing speeding, driving drunk, driving aggressively for conditions, being drowsy, and being distracted as key contributors to accidents. “The hope—though at this point it is a hope—is that automation



can significantly reduce these kinds of crashes without introducing significant new sources of errors.”

However, should an unavoidable crash situation arise, a driverless car’s method of seeing and identifying potential objects or hazards is different and less precise than the human eye-brain connection, which likely will introduce moral dilemmas with respect to how an autonomous vehicle should react, according to Patrick Lin, director of the Ethics + Emerging Sciences Group at California Polytechnic State University, San Luis Obispo. Lin says the vision technology used in driverless cars still has a long way to go before it will be morally acceptable for use.

“We take our sight and ability to distinguish between objects for granted, but it’s still very difficult for a computer to recognize an object as that object,” Lin says, noting that today’s light-detection and ranging (LIDAR)-based machine-vision systems used on autonomous cars simply “see” numerical values related to the brightness of each pixel of the image being scanned, and then infer what the object might be.

Lin says with specific training, it eventually will be technically feasible to

create a system that can recognize baby strollers, shopping carts, plastic bags, and actual boulders, though today’s vision systems are only able to make very basic distinctions, such as distinguishing pedestrians from bicyclists.

“Many of the challenging scenarios that an autonomous car may confront could depend on these distinctions, but many others are problematic exactly because there’s uncertainty about what an object is or how many people are involved in a possible crash scenario,” Lin says. “As sensors and computing technology improves, we can’t point to a lack of capability as a way to avoid the responsibility of making an informed ethical decision.”

Assuming eventually these technical challenges will be overcome, it will be possible to encode and execute instructions to direct the car how to respond to a sudden or unexpected event. However, the most difficult part is deciding what that response should be, given that in the event of an impending or unavoidable accident, drivers are usually faced with a choice of at least two less-than-ideal outcomes.

For example, in the event of an unavoidable crash, does the car’s pro-

gramming simply choose the outcome that likely will result in the greatest potential for safety of the driver and its occupants, or does it choose an option where the least amount of harm is done to any of those involved in an accident, such as having the car hit a telephone pole with the potential to cause the driver a relatively minor injury, instead of striking a (relatively) defenseless pedestrian, bicyclist, or motorcycle rider, if the driver is less likely to be injured?

The answer is not yet clear, though the moral decisions are unlikely to reside with users, given their natural propensity to protect themselves against even minor injuries, often at the expense of others, Lin says.

"This is a giant task in front of the industry," Lin says. "It's not at all clear who gets to decide these rules. In a democracy, it's not unreasonable to think that society should have input into this design decision, but good luck in arriving at any consensus or even an informed decision."

One potential solution would be the creation and use of institutional review boards, which would compel autonomous vehicle manufacturers to provide potential crash scenarios, explain what its vehicles' capabilities or responses to those scenarios would be, and document and explain why programmers made those choices.

Jonathan Handel, a computer scientist turned lawyer, explains that rather than try to come up with hard-and-fast rules now, when driverless cars have yet to interact on public roads outside of tightly controlled testing runs, these review boards would provide a process to allow manufacturers, lawyers, ethicists, and government entities to work through these nascent, yet important, ethical decisions.

"I propose ethics review boards, or institutional review boards," Handel says. "I don't think that we're at a place in this technology, nor do I think we will be in the first few years of it [being used], that there would be an obvious, one good answer to all these questions. For the ethics issue, I think we need a procedural answer, not a substantive one."

He adds, "Eventually, consensus may emerge organically on various issues, which could then be reflected in regulations or legislation."

Given the near-infinite number of

potential situations that can result in an accident, it would seem resolving these issues before driverless cars hit the road en masse would be the only ethical way to proceed. Not so, say technologists, noting unresolved ethical issues have always been in play with automobiles.

"In some ways, there are ethical issues in today's products," Smith says. "If you choose [to drive] an SUV, you are putting pedestrians at greater risk [of injury], even though you would believe yourself to be safer inside, whether or not that's actually true."



Further, a high degree of automation is already present in vehicles on the road today. Adaptive cruise control, lane-keeping assist technology, and even self-parking technology is featured on many vehicles, with no specific regulatory or ethical guidelines for use.

In all likelihood, Google, Volkswagen, Mercedes, and the handful of other major auto manufacturers that are pressing ahead with driverless cars are unlikely to wait for ethical issues to be fully resolved. It is likely basic tenets of safe vehicle operation will be programmed, such as directing the car to slow down and take the energy out of a potential crash, avoiding "soft" targets such as pedestrians, cyclists, or other smaller objects, and selecting appropriate trade-offs (choosing the collision path that might result in the least severe injury to all parties involved in an accident) to be employed.

Another option to potentially deal with moral issues would be to cede control back to the driver during periods of congestion or treacherous conditions, so the machine is not required to make moral decisions. However, this approach is flawed: emergency situations can occur at any time; humans are usually unable to respond to a situation fast enough after being disengaged and, in the end, machines are likely able to respond faster and more accurately than humans to emergency situations.

"When a robot car needs its human driver to quickly retake the wheel, we're going to see new problems in the time it takes that driver to regain enough

situational awareness to operate the car safely," Lin explains. "Studies have shown a lag-time anywhere from a couple seconds to more than 30 seconds—for instance, if the driver was dozing off—while emergency situations could occur in split-seconds."

This is why Google and others have been pressing ahead for a fully autonomous vehicle, though it is likely such a vehicle will not be street-ready for at least five years, and probably more. The navigation and control technology has yet to be perfected (today's driverless cars tooling around the roads of California and other future-minded states still are unable to perform well in inclement weather, such as rain, snow, and sleet, and nearly every inch of the roads used for testing have been mapped.)

Says Lin, "legal and ethical challenges, as well as technology limitations, are all part of the reason why [driverless cars] are not more numerous or advanced yet," adding that industry predictions for seeing autonomous vehicles on the road vary widely, from this year to 2020 and beyond.

As such, it appears there is time for manufacturers to work through the ethical issues prior to driverless cars hitting the road. Furthermore, assuming the technological solutions can provide enhanced awareness and safety, the number of situations that require a moral decision to be made will become increasingly infrequent.

"If the safety issues are handled properly, the ethics issues will hopefully be rarer," says Handel. ■

Further Reading

Thierer, A., and Hagemann, R., *Removing Roadblocks to Intelligent Vehicles and Driverless Cars*, Mercatus Working Paper, September 2014, <http://bit.ly/1CohoV8>

Levinson, J., Askeland, J., Becker, J., and Dolson, J., *Towards fully autonomous driving: Systems and algorithms*, *Intelligent Vehicles Symposium IV* (2011), <http://bit.ly/1BKd5A4>

Ensor, J., *Roadtesting Google's new driverless car*, *The Telegraph*, <http://bit.ly/1x8Vgfb>

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in Lynbrook, NY.

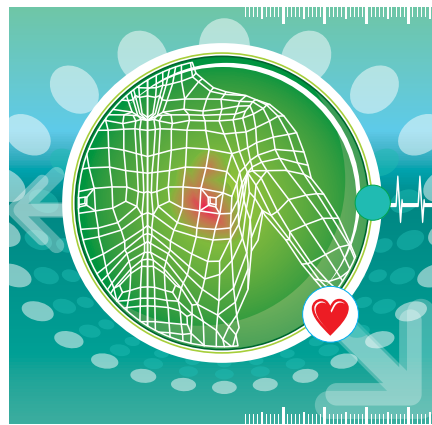
© 2015 ACM 0001-0782/15/08 \$15.00

Privacy and Security Security for Mobile and Cloud Frontiers in Healthcare

Designers and developers of healthcare information technologies must address preexisting security vulnerabilities and undiagnosed future threats.

I FEAR THE day when your security requirement kills one of my patients,” said a medical practitioner to the security professionals proposing improved security for the clinical information system. Every security professional is familiar with the challenge of deploying strong security practices around enterprise information systems, and the skepticism of well-intentioned yet uncooperative stakeholders. At the same time, security solutions can be cumbersome and may actually affect patient outcomes.

Information technology (IT) has great potential to improve healthcare, promising increased access, increased quality, and reduced expenses. In pursuing these opportunities, many healthcare organizations are increasing their use of mobile devices, cloud services, and Electronic Health Records (EHRs). Insurance plans and accountable-care organizations encourage regular or even continu-



ous patient monitoring. Yet *The Washington Post* found healthcare IT to be vulnerable and healthcare organizations lagging behind in addressing known problems.⁹ Recent breaches at two major health insurance companies^{1,7} underscore this point: the healthcare industry moves toward automation and online records, yet falls behind when addressing security and privacy, ranking below retail in terms of cybersecurity.³

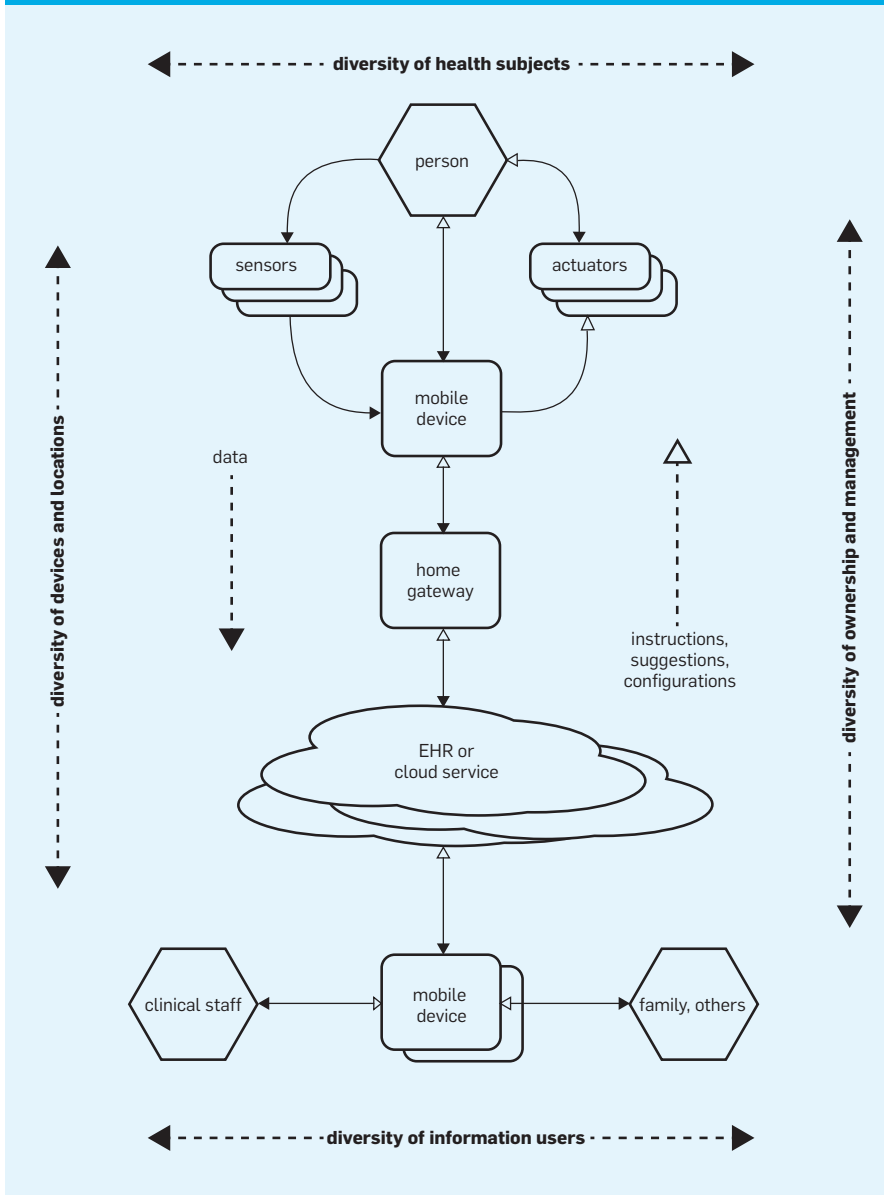
The benefits of healthcare IT will be elusive if its security challenges are not adequately addressed. Security remains one of the most important concerns in a recent survey of the health and mHealth sectors,¹² and research has illustrated the risks incurred by cyberattacks on medical devices such as pacemakers.⁵ More than two-thirds (69%) of respondents say their organization’s IT security does not meet expectations for FDA-approved medical devices.⁶

Privacy protection is also critical for healthcare IT; although this column focuses on security, it should be noted that many security breaches lead to disclosure of personal information and thus an impact on patient privacy.

Critical Research Challenges

The accompanying figure shows the complex trust relationships involved. Those who use medical information are diverse: families, clinicians, researchers, insurers, and employers are some

The complex trust relationships involved in healthcare information technologies.



examples. Those who provide information are also diverse: traditional patients, healthy athletes, children, the elderly, and so forth. The mobile devices and cloud systems are also diverse and are often managed by multiple organizations. The result is a complex mix of trust relationships with implications both for technology and the social, economic, and regulatory environment in which the technology operates.

Designers and developers of healthcare information technologies can help by designing security into all devices, apps, and systems, and by developing policies and practices that recognize the rights of individuals regarding information collected on them: where it will be stored, how it will be used,

and whom will have access. Researchers should develop new methods for authentication, identification, data anonymization, software assurance, device and system management—and human factors should play a critical role in all of these methods. Some of the most-critical research challenges are described here.

Usable authentication tools. Health IT presents many demanding problems for users in authenticating themselves to systems. Traditional authentication mechanisms like passwords can disrupt workflow and interfere with the primary mission of patient care. New authentication mechanisms must blend into the clinical workspace, recognize that staff often wear gloves and

masks (obviating solutions based on face and fingerprint recognition), and work with smartphones, tablets, desktops, and laptops.

EHR systems should not arbitrarily limit clinical staff from viewing an entire record—denying access in an emergency situation may lead to delayed care or even death. However, “break-the-glass” provisions of many EHRs to provide emergency access to patient records make more information available than necessary for care. Break-the-glass mechanisms should expose patient records in stages to provide needed information without providing too much information, and trigger automated and organizational audit mechanisms.

Patients are increasingly asked to use (or wear) mHealth technology outside the clinical setting, but might not want mHealth data to reveal detailed activities. They might want to suspend reporting for periods of time, or block systems from storing or sharing data that is not directly relevant to treating their condition. Mechanisms should separate data *collection*, *analysis*, and *presentation* to limit data that travels outside the patient’s trust circle. These mechanisms should be easy to understand and use, indicating how data may be collected, stored, and shared. Fine-grained consent descriptions should interoperate across mHealth systems and EHRs, and travel with data that flows from one system to another. The foundation of any privacy-supporting solution is a secure system with strong mechanisms for identifying and authenticating users.

The declining cost of gene sequencing enables a new generation of precision medicine.¹¹ Although this technology has great promise, basic issues have yet to be handled: how patients should access their own genomic information, how they control sharing with health professionals, and how best to provide “direct to consumer” services like support for genealogy explorations.⁸

Trustworthy control of medical devices. Today’s sophisticated medical devices like infusion pumps and vital-sign monitors are increasingly networked (possibly via the Internet) and run safety-critical software. Network-capable medical devices may have cyber-security vulnerabilities that can

have implications for patient safety. Medical devices must contain defenses against today's known vulnerabilities and tomorrow's anticipated threats.²

Medical devices must defend against conventional malware that attacks their outdated operating systems. For example, Conficker and bot-net malware can break into unmaintained systems easily; old operating systems provide large reservoirs for the Conficker worm, and medical devices can have long product life cycles that persist with outdated operating-system software. MRI machines running Windows 95, pacemaker programmers recently upgraded from OS/2 to Windows XP, and pharmaceutical compounders running Windows XP Embedded have been noted.

Medical devices must also withstand threats that match the future product life cycle, but it is difficult to secure a device for 20 years. The 1995 desktop computer could not withstand today's threats of spam, malware, drive-by-downloads, and phishing attacks. It is difficult to design medical devices for evolving threats. According to the Veterans Administration, modern malware can enter via USB drives used by contractors upgrading medical-device software. Better methods are needed to engineer secure software and ensure the correct software is running. Improvements are needed for detecting attempted network attacks (wired or wireless), and for dealing with attacks in progress without compromising patient safety. Solutions aimed at desktop computers and Internet servers might not work for medical devices. For more on this topic, see the recent *Communications* article by Sametinger et al.¹⁰

Trust through accountability. Health IT provides a foundation for diagnosis, treatment, and other medical decision making. This foundation must be both *dependable* and *trustworthy*. Technical security is essential, but trust also critically depends on social, organizational, and legal frameworks behind the technology. Health IT must be *accountable*, which means people and organizations must be held responsible for the ways the systems are used. Systems configured to provide access to many must be backed by responsible organiza-

The benefits of healthcare IT will be elusive if its security challenges are not adequately addressed.

tions that determine who has access and when. "Break-glass" mechanisms must be guided by protocols about who can break the glass, and for what purpose.

Audit logs of all health IT systems are needed to monitor for buggy or inappropriate behavior, and to support post-event analysis as well as the development of proper access controls.⁴ There has been considerable study of audit logs and accountability for hospital patient records, but mobile systems and devices also need rigorous auditing. Automated analysis of audit logs in medical systems would be useful, as would be the ability to detect anomalies (such as staff members looking at rarely examined records or device settings changed by a person not normally given access to the device). Access restrictions should be imposed according to workflow data and/or models trained via machine learning to diminish reliance on post-hoc accountability. There are many research opportunities in this space.

Conclusion

The research community must address many fundamental and practical challenges to enable healthcare IT to achieve the level of security essential for widespread adoption and successful deployment. For doctors and other caregivers to embrace more secure solutions, they need to be usable and fit within their clinical workflow. For patients and family members to accept these technologies, they need to be comfortable with the privacy of their personal information and able to effectively use the security solutions that support those privacy mechanisms. We call on the research community to tackle these challenges with us. 

References

1. Eastwood, B. Premiera says data breach may affect 11 million consumers. *FierceMobileHealthcare* (Mar. 18, 2015); <http://www.fiercehealthit.com/story/premera-says-data-breach-may-affect-11-million-consumers/2015-03-18>.
2. Fu, K. Trustworthy medical device software. In *Public Health Effectiveness of the FDA 510(k) Clearance Process: Measuring Postmarket Performance and Other Select Topics*. IOM (Institute of Medicine) Workshop Report, National Academies Press, Washington, D.C., July 2011; <https://spqr.eecs.umich.edu/papers/fu-trustworthy-medical-device-software-IOM11.pdf>.
3. Gagliardi, N. Healthcare cybersecurity worse than retail: BitSight. (May 28, 2014); <http://www.zdnet.com/article/healthcare-cybersecurity-worse-than-retail-bitsight/>.
4. Gunter, C.A., Liebovitz, D.M., and Malin, B. Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE Security & Privacy* 9, 5 (Sept./Oct. 2011); DOI 10.1109/MSP.2011.72.
5. Halperin, D. et al. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*. IEEE Press (May 2008), 129–142; DOI: 10.1109/SP.2008.31.
6. Ponemon Institute. Third annual benchmark study on patient privacy and data security (Dec. 2012); http://www.ponemon.org/local/upload/file/Third_Annual_Study_Patient_Privacy_FINAL5.pdf.
7. Millions of Anthem customers targeted in cyberattack. *New York Times* (Feb. 5, 2015); <http://www.nytimes.com/2015/02/05/business/hackers-breached-data-of-millions-insurer-says.html>.
8. Naveed, M. et al. Privacy in the genomic era. *ACM Comput. Surv.* 48, 1, Article 6 (July 2015); DOI: <http://dx.doi.org/10.1145.2767007>.
9. O'Harrow, Jr., R. Health-care sector vulnerable to hackers, researchers say. *Washington Post* (Dec. 2012); http://articles.washingtonpost.com/2012-12-25/news/36015727_1_health-care-medical-devices-patient-care.
10. Sametinger, J., Rozenblit, J., Lysecky, R., and Ott, P. Security challenges for medical devices. *Commun. ACM* 58, 4 (Apr. 2015), 74–82; DOI 10.1145/2667218.
11. White House. FACT SHEET: President Obama's Precision Medicine Initiative (Jan. 30, 2015); <https://www.whitehouse.gov/the-press-office/2015/01/30/fact-sheet-president-obama-s-precision-medicine-initiative>.
12. Whittaker, R. Issues in mHealth: Findings from key informant interviews. *Journal of Medical Internet Research* 14, 5 (May 2012); DOI 10.2196/jmir.1989.

David Kotz (kotz@cs.dartmouth.edu) is a professor of computer science at Dartmouth College, principal investigator of the NSF-funded Trustworthy Health and Wellness (THaW.org) project, and former director of the Institute for Security, Technology, and Society (ISTS).

Kevin Fu (kevinfu@umich.edu) is an associate professor of electrical engineering and computer science at the University of Michigan, a member of the NIST Information Security and Privacy Advisory Board, a member of the ACM Committee of Computers and Public Policy, former ORISE Fellow at the FDA, and director of the Archimedes Center for Medical Device Security.

Carl Gunter (cgunter@illinois.edu) is a professor of computer science, a professor in the College of Medicine, and the director of the Illinois Security Lab and the Health Information Technology Center at the University of Illinois, Urbana.

Avi Rubin (rubin@jhu.edu) is a professor of computer science and technical director of the Information Security Institute at Johns Hopkins University, and principal investigator of one of the first NSF CyberTrust centers (on e-voting).

This research program is supported by a collaborative award from the National Science Foundation (NSF award numbers CNS-1329686, 1329737, 1330142, and 1330491). The views and conclusions contained in this material are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF. Any mention of specific companies or products does not imply any endorsement by the authors or by the NSF.

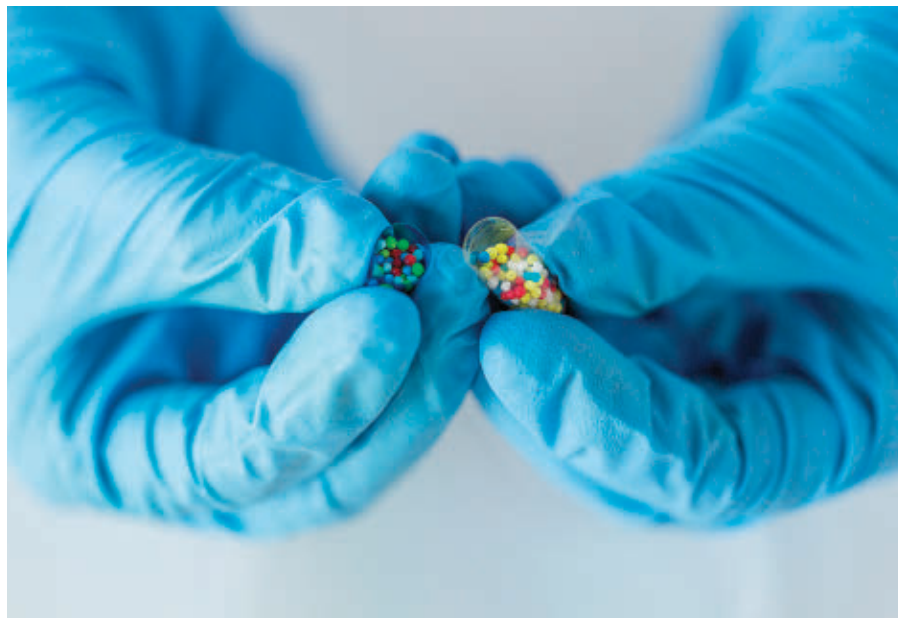
Copyright held by authors.

Economic and Business Dimensions Permissionless Innovation

Seeking a better approach to pharmaceutical research and development.

PERMISSIONLESS INNOVATION," IS the freedom to explore new technologies or businesses without seeking prior approval.¹⁴ It has already produced an explosion of goods and services in the IT industry.¹⁴ Vint Cerf, a father of the Internet, invokes it when he argues the Web must remain open.⁴ It is cousin to the end-to-end principle of placing application-specific functions at end points, where others can build, rather than in the core.¹¹ It improves efficiency and moves innovation closer to people with ideas.³ Hundreds of thousands of iOS and Android apps were not created by Apple or Google, but by permissionless innovation made possible by published APIs (application programming interfaces) and resulting market evolution. It facilitates experimentation in parallel: actors launch their own experiments without depending on the results of others. Permissionless innovation greatly increases the speed of invention and allows the ecosystem to provide ideas its system designers never had.¹

The pharmaceutical industry could benefit from this approach. A successful new drug can cost upward of \$800 million.^{5,6} Uncertainty about winning patent races means firms race to secure intellectual property rights. Competitors need patent shears to trim patent thickets.^{5,6} Yet the awful expense of maintaining IP creates abandonment



problems due to the lag between costs and revenues. Innovation slows⁸ while, ironically, only 8% of pharma firms measure the value of their orphan patents.²

Industries outside the IT sector have shown the benefit permissionless innovation. One example is the Canadian firm Goldcorp, written off by analysts due to debt and high production costs.¹² After decades of use, Goldcorp's Red Lake mine was underperforming. In-house geologists could not locate further gold deposits. When Goldcorp CEO Rob McEwan heard an MIT presentation on the Linux operating sys-

tem, which allows anyone to access the code, he saw an opportunity. He gave away Goldcorp geological data going back to 1948 and challenged the world to find gold in his data. The world did. New models for discerning ore, visualizing veins, and improving extraction were submitted, along with 110 sites to look for gold at Red Lake—half of them unknown to Goldcorp. Stock purchased a year before and sold a year after McEwan's decision tripled in value.

Goldcorp gave permission and incentives. It paid out \$565 million, its highest payout to two collaborating

Australian teams that had never visited the mines. But Goldcorp retained an economic complement, deeds to the 55 acres of Red Lake mines. Any increases in value applied directly to Goldcorp's assets. Permissionless innovation opened new insights that complemented Goldcorp's assets in the same way that Apple and Google opening their APIs added value to their operating systems. McEwan did not give away the business, he gave away permission to help find gold on his property.

Pharmaceutical firms can do the same with their tapped-out mines. FDA review effectively handles Type 1 errors, or false positives, such that unsafe, ineffective drugs seldom reach market. However, Type 2 errors—false negatives with more promise than their owners realize—can be the abandoned veins of hidden gold. Viagra could have been one of these. It showed limited promise as a hypertension drug, but the men in the treatment group refused to return their samples at the end of human trials, and treatment for erectile dysfunction has never the same. Similarly, Eli Lilly discovered the antibiotic Cubicin in the 1980s but abandoned it because it was too toxic to patients. After licensing, a small unrelated firm developed effective methods to poison bacteria without poisoning people, and Cubicin became a front-line treatment against antibiotic-resistant or “super bug” staphylococcus—the most successful intravenous antibiotic ever launched in the U.S.⁷ The value of Viagra and Cubicin came from the willingness to be innovative in ways not originally foreseen, to open innovation to outsiders and get value from what would otherwise be false-negatives.

The explosion of innovation in the IT industry comes from modularization of complex technologies and APIs to enable outsiders to access technologies without asking for permission. Modular technologies can allow firms to benefit from innovations of others by creating a platform that connects these elements together in useful, distinctive ways. They separate the core technology of a firm from extensions outside a firm as a governance model for innovation. They direct external parties to make contributions that are invited and welcomed. They help the platform owner benefit from third-party innova-

tion by clarifying rules without requiring face-to-face interaction in advance.

New emerging rules apply to platforms that promote innovation.⁹ Crafting technology to protect closed parts benefits from open access like APIs so others can use the technology without knowing what is inside it. Donating technological assets to a commons or providing subsidy for open technology (APIs and SDKs or software development kits) can retain ownership over complementary assets one wishes to monetize.¹³ Encouraging entry into the open part of the technology lets a thousand flowers bloom and does not prevent selectively pricing access to the closed part of the platform. The open source dictum “with many eyes, all bugs are shallow”¹⁰ has a counterpart in open innovation: with open APIs more ingenuity is possible. The platform owner can observe external contributions that become popular in the market, but they must not grab too much of this new wealth for themselves. Third-party extensions make the platform more valuable for customers and more attractive for future innovators to drive even further innovation. This happens only if one does not confiscate what other people build. Successful platforms partner with innovators to share the wealth, buy them at a fair price, or grant them a grace period to collect their rewards. For example, enterprise software firm SAP outlines its future projects in a public two-year roadmap to let external parties know where to build, and what to avoid, in order to pursue opportunities in the platform's ecosystem.⁹

So how do we apply this to pharmaceuticals, which are complex and where partitioning is challenging? Separating the process into open and closed domains and stimulating permissionless innovation in the open arena is a start. Like Goldcorp, a pharma company can provide open access to preclinical and clinical data on a particular compound, and award prizes for third parties who determine the best diseases to target, or which variations to pursue. Assays to proxy therapeutic benefit might be shared widely, and compounds with “hits” (signs of positive activity) can become a part of negotiation for development. Computational models of particular compounds can be shared.

Calendar of Events

August 7–9

HPG '15: High Performance Graphics,
Los Angeles, CA
Contact: Steven Molnar
Email: molnar@nvidia.com

August 7–9

SCA '15: The ACM SIGGRAPH/
Eurographics Symposium on
Computer Animation,
Los Angeles, CA,
Contact: Jernej Barbic,
Email: jernej.barbic@gmail.com

August 8

DigiPro '15: The Digital
Production Symposium,
Los Angeles, CA,
Sponsored: ACM/SIG,
Contact: Eric Enderton
Email: acme@enderton.org

August 8–9

SUI '15: Symposium on
Spatial User Interaction,
Los Angeles, CA,
Co-Sponsored: ACM/SIG,
Contact: Amy C. Banic
Email: amyulinski@gmail.com

August 9–13

ICER '15: International
Computing Education
Research Conference,
Omaha, NE,
Sponsored: ACM/SIG,
Contact: Brian Dorn,
Email: bdorn@unomaha.edu

August 9–13

SIGGRAPH '15: Special Interest
Group on Computer Graphics
and Interactive Techniques
Conference,
Los Angeles, CA,
Sponsored: ACM/SIG,
Contact: Marc Barr
Email: marc.barr@mtsu.edu

August 17–21

SIGCOMM '15: ACM SIGCOMM
2015 Conference,
London, UK,
Sponsored: ACM/SIG,
Contact: Steve Uhlig
Email: steve.uhlig@gmail.com

August 25–28

MobileHCI '15: 17th International
Conference on Human-
Computer Interaction with
Mobile Devices and Services,
Copenhagen, Denmark,
Sponsored: ACM/SIG,
Contact: Enrico Rukzio,
Email: enrico@rukzio.de

Call for Nominations for ACM General Election

The ACM Nominating Committee is preparing to nominate candidates for the officers of ACM: **President, Vice-President, Secretary/Treasurer; and five Members at Large.**

Suggestions for candidates are solicited. Names should be sent by **November 5, 2015** to the Nominating Committee Chair, c/o Pat Ryan, Chief Operating Officer, ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA.

With each recommendation, please include background information and names of individuals the Nominating Committee can contact for additional information if necessary.

Vinton G. Cerf is the Chair of the Nominating Committee, and the members are Michel Beaudouin-Lafon, Jennifer Chayes, P.J. Narayanan, and Douglas Terry.



Think of permissionless innovation as a complement to traditional research and development.

Companies still can retain the patents. Drug companies already partition rights to IP for research, field of use, geographic region, or preset conditions such as royalties. A “part open, part closed” governance structure permits pharma companies to invite others to examine compounds and data and run experiments without prior permission. We already see this in practice when clinicians use “off label” drugs to treat unmet medical needs. Drugs approved by the FDA for one treatment sometimes work for another.

If a pharma company controls the platform, why would others play? Pharma companies might keep the best drugs closed, opening only the least attractive. They can use platforms that are too closed and do not attract permissionless innovative activity. We argue that third parties are sophisticated enough to differentiate between opportunity and lip service. Why presume that pharma companies have all the ideas? They do not know everything outside their core technologies and markets. Third parties with specialized knowledge can find opportunities pharma companies miss, just like others found veins of gold missed by Goldcorp. Specialty companies can seek niche markets unattractive to large companies. Mission-oriented nonprofits with humanitarian motives can lift economic restrictions when looking for therapies. In permissionless innovation, all have a role to play.

Turning things around, why would pharma companies willingly open technology, data, and IP? Pharma companies are increasingly specialized by disease, but their compounds might have benefits in other areas. Epilepsy drugs, for example, are prescribed for

bipolar disorder, depression, and neuropathic pain. Drugs developed for one use are used for different conditions, as with Viagra. Drug development costs and the narrowing market focus create areas where permissionless innovation makes sense. Independent researchers are already looking for new diseases to target, using data and compounds of pharma companies.

Think of permissionless innovation as a complement to traditional research and development. Everything done to discover new drugs can still be done, but permissionless innovation adds access and incentive so third parties contribute to solving challenges that lie beyond the capacity of traditional pharma firms. Ecosystems can rejuvenate research and development. Through better governance, open innovation, and strategic platform management, smart firms can have thousands of partners innovate on their IP but on their terms and without prior permission. The future is open. 

References

1. Andreesen, M. The three kinds of platform you meet on the Internet (Sept. 16, 2007); http://pmarchive.com/three_kinds_of_platforms_you_meet_on_the_internet.html.
2. Arora, A., Fosfuri, A., and Gambardella, A. *Markets for Technology: The Economics of Innovation and Corporate Strategy*. MIT Press, 2001.
3. Baldwin, C. and Clark, K. *Design Rules*. MIT Press, Cambridge, MA, 2001.
4. Cerf, V. Keep the Internet open. *New York Times* (May 24, 2012); <http://www.nytimes.com/2012/05/25/opinion/keep-the-internet-open.html>.
5. Chesbrough, H. *Open Innovation: The New Imperative for Creating and Profiting from Technology*. HBS Press, Boston, MA, 2003.
6. Chesbrough, H. Why companies should have open business models. *MIT Sloan Management Review* 48, 2 (2012).
7. Chesbrough, H.W. and Chen, E.L. Recovering abandoned compounds through expanded external IP licensing. *California Management Review* 55, 4 (Apr. 2013), 83–101.
8. Hargreaves, I. Digital opportunity: A review of intellectual property and growth. An independent report. 2001.
9. Parker, G. and Van Alstyne, M. Innovation, openness, and platform control. *Mimeo: Boston University* (2015). <http://ssrn.com/abstract=1079712>.
10. Raymond, E. *The Cathedral and the Bazaar*. O'Reilly, 1999.
11. Saltzer, J.H., Reed, D.P., and Clark, D.D. End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)* 2, 4 (1984), 277–288.
12. Tapscott, D. and Williams, D.A. Innovation in the age of mass collaboration. *Business Week* (Feb. 1, 2007).
13. Teece, D. Profiting from innovation. *Research Policy*, 1986.
14. Thierer, A. *Permissionless Innovation: The Continuing Case for Comprehensive Technological Freedom*. Mercatus Center at George Mason University, 2014.

Henry Chesbrough (chesbrou@haas.berkeley.edu) is a professor at UC Berkeley's Haas Business School. He is the author of *Open Innovation*, and five other innovation books.

Marshall Van Alstyne (mva@bu.edu) is a professor of information economics at Boston University, a visiting professor at the MIT Initiative on the Digital Economy, and coauthor of *Platform Revolution*; Twitter: InfoEcon.

Copyright held by authors.



Kode Vicious Hickory Dickory Doc

On null encryption and automated documentation.

Dear KV,

While reviewing some encryption code in our product, I came across an option that allowed for null encryption. This means the encryption could be turned on, but the data would never be encrypted or decrypted. It would always be stored “in the clear.” I removed the option from our latest source tree because I figured we did not want an unsuspecting user to turn on encryption but still have data stored in the clear. One of the other programmers on my team reviewed the potential change and blocked me from committing it, saying the null code could be used for testing. I disagreed with her, since I think the risk of accidentally using the code is more important than a simple test. Which of us is right?

NULL for Naught

Dear NULL,

I hope you are not surprised to hear me say that she who blocked your commit is right. I have written quite a bit about the importance of testing and I believe that crypto systems are critical enough to require extra attention. In fact, there is an important role that a null encryption option can play in testing a crypto system.

Most systems that work with cryptography are not single programs, but are actually frameworks into which differing cryptographic algorithms can be placed, either at build or run-time. Cryptographic algorithms are also well known for requiring a great deal of processor resources, so much



so that specialized chips and CPU instructions have been produced to increase the speed of cryptographic operations. If you have a crypto framework and it does not have a null operation, one that takes little or no time to complete, how do you measure the overhead introduced by the framework itself? I understand that establishing a baseline measurement is not common practice in performance analysis, an understanding I have come to while banging my fist on my desk and screaming obscenities. I often think that programmers should not just be given offices instead of cubicles, but padded cells. Think of

how much the company would save on medical bills if everyone had a cushioned wall to bang their heads against, instead of those cheap, pressboard desks that crack so easily.

Having a set of null crypto methods allows you and your team to test two parts of your system in near isolation. Make a change to the framework and you can determine if that has sped up or slowed down the framework overall. Add in a real set of cryptographic operations, and you will then be able to measure the effect the change has on the end user. You may be surprised to find that your change to the framework did not speed up the system overall, as

it may be the overhead induced by the framework is quite small. But you cannot find this out if you remove the null crypto algorithm.

More broadly, any framework needs to be tested as much as it can be in the absence of the operations that are embedded within it. Comparing the performance of network sockets on a dedicated loopback interface, which removes all of the vagaries of hardware, can help establish a baseline showing the overhead of the network protocol code itself. A null disk can show the overhead present in file-system code. Replacing database calls with simple functions to throw away data and return static answers to queries will show you how much overhead there is in your Web and database framework.

Far too often we try to optimize systems without sufficiently breaking them down or separating out the parts. Complex systems give rise to complex measurements, and if you cannot reason about the constituent parts, you definitely cannot reason about the whole, and anyone who claims they can, is lying to you.

KV

Dear KV,

What do you think of systems such as Doxygen that generate documentation from code? Can they replace handwritten documentation in a project?

Dickering with Docs

Dear Dickering,

I am not quite sure what you mean by “handwritten” documentation. Unless you have some sort of fancy mental interface to your computer that I have not yet heard of, any documentation, whether in code or elsewhere is handwritten or at least typed by hand. I believe what you are actually asking is if systems that can parse code and extract documentation are helpful, to which my answer is, “Yes, but ...”

Any sort of documentation extraction system has to have something to work with to start. If you believe that extracting all of the function calls and parameters from a piece of code is sufficient to be called documentation, then you are dead wrong, but, unfortu-

For code to be considered documented there must be some set of expository words associated with it.

nately, you would not be alone in your beliefs. Alas, having beliefs in common with others does not make those beliefs right. What you will get from Doxygen on the typically, uncommented, code base is not even worth the term “API guide,” it is actually the equivalent of running a fancy grep over the code and piping that to a text formatting system such as TeX or troff.

For code to be considered documented there must be some set of expository words associated with it. Function and variable names, descriptive as they might be, rarely explain the important concepts hiding in the code, such as, “What does this damnable thing actually do?” Many programmers claim their code is self-documenting, but, in point of fact, self-documented code is so rare that I am more hopeful of seeing a unicorn giving a ride to a mantichore on the way to a bar. The claim of self-documenting code is simply a cover up for laziness. At this point, most programmers have nice keyboards and should be able to type at 40–60 words per minute, some of those words can easily be spared for actual documentation. It is not like we are typing on ancient line-printing terminals.

The advantage you get from a system like Doxygen is that it provides a consistent framework in which to write the documentation. Setting off the expository text from the code is simple and easy and this helps in encouraging people to comment their code. The next step is to convince people to ensure their code matches the comments. Stale comments are sometimes worse than none at all because they can misdirect you when looking for a bug in the code. “But it

says it does X!,” is not what you want to hear yourself screaming after hours of staring at a piece of code and its concomitant comment.

Even with a semiautomatic documentation extraction system, you still need to write documentation, as an API guide is not a manual, even for the lowest level of software. How the API’s documentation comes together to form a total system and how it should and should not be used are two important features in good documentation and are the things that are lacking in the poorer kind. Once upon a time I worked for a company whose product was relatively low level and technical. We had automatic documentation extraction, which is a wonderful first step, but we also had an excellent documentation team. That team took the raw material extracted from the code and then extracted, sometimes gently and sometimes not so gently, the requisite information from the company’s developers so they could not only edit the API guide, but then write the relevant higher-level documentation that made the product actually usable to those who had not written it.

Yes, automatic documentation extraction is a benefit, but it is not the entire solution to the problem. Good documentation requires tools and processes that are followed rigorously in order to produce something of value both to those who produced it and to those who have to consume it.

KV

Related articles on queue.acm.org

API: Design Matters

Michi Henning

<http://queue.acm.org/detail.cfm?id=1255422>

Microsoft’s Protocol Documentation Program: Interoperability Testing at Scale

A Discussion with Nico Kicillof,

Wolfgang Grieskamp, and Bob Binder

<http://queue.acm.org/detail.cfm?id=1996412>

Kode Vicious vs. Mothra!

George Neville-Neil

<http://queue.acm.org/detail.cfm?id=1066061>

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.

Education

Understanding the U.S. Domestic Computer Science Ph.D. Pipeline

Two studies provide insights into how to increase the number of domestic doctoral students in U.S. computer science programs.

RECRUITING DOMESTIC STUDENTS (U.S. citizens and permanent residents) into computer science Ph.D. programs in the U.S. is a challenge for most departments, and the health of the “domestic Ph.D. pipeline” is of concern to universities, companies, government agencies, and federal research labs. In this column, we present results from two studies on the domestic research pipeline carried out by CRA-E, the Education Committee of the Computing Research Association. The first study examined the baccalaureate origins of domestic Ph.D. recipients; the second study analyzed applications, acceptances, and matriculation rates to 14 doctoral programs. Informed by findings from these studies, we also present recommendations we believe can strengthen the domestic Ph.D. pipeline.

While international students are—and will remain—crucial to the vitality of U.S. doctoral programs, an increasing number of these graduates return to their countries of origin for competitive job opportunities. The demand for new computer science Ph.D.’s is high. Currently, approximately 1,600 computer science Ph.D.’s are awarded



The seven producer groups.

Producer Group	Number of Schools	Explanation
TOP4	4	Top 4 CS departments (CMU, MIT, Stanford, UC Berkeley)
TOP25-4	21	21 CS departments ranked in the top 25 by <i>USNWR</i> and excluding the Top 4 CS departments
RU/VH-25	83	Universities with a Carnegie classification of "Very High" research excluding the Top 25 CS departments in <i>USNWR</i>
RU/H	99	Universities whose Carnegie classification is "High" research
Master's	724	Schools with a Carnegie classification of Master's institution
TOP25LA+	27	Schools ranked in the Top 25 Liberal Arts Colleges by <i>USNWR</i> plus Olin and Rose-Hulman
BAC-25LA	246	Schools with a Carnegie classification of Baccalaureate Arts and Sciences four-year college excluding the TOP 25 LA

each year from U.S. institutions, with approximately 55% of these Ph.D.'s hired by companies and federal research labs. Of the 1,600 Ph.D.'s, approximately 47% go to domestic students.¹ Lucrative salaries and compelling jobs for new college graduates add to the challenge of encouraging domestic students to pursue a Ph.D.

A first step in understanding the pipeline of domestic students to Ph.D. programs is to examine baccalaureate origins of domestic doctoral students.⁴ This is the focus of the first of our studies, which found a relatively small number of colleges and universities are the undergraduate schools of origin for most domestic Ph.D. students. Specifically, for the years 2000–2010, just 12 schools accounted for 20% of the approximately 5,500

domestic bachelor's graduates who received a Ph.D. in computer science and 54 schools were the undergraduate origins for approximately 50% of domestic Ph.D.'s. Approximately 730 other schools were the origins of the other 50%; on average, these schools had less than one graduate every three years receive a computer science Ph.D.

To better understand the domestic student pipeline and ultimately make recommendations for how to improve it, we worked with 14 U.S. computer science Ph.D. programs that provided graduate admissions records during the period 2007–2013. This data, a total of 7,032 graduate admissions records from domestic applicants, formed the basis of our second study. The 14 departments ranked 5 to 70 (out of 177 graduate

programs ranked) in the 2014 *U.S. News and World Report (USNWR)* ranking of graduate programs in computer science⁶ and during the 2007–2013 period they produced about 19% of all computer science Ph.D.'s in the U.S. For those 14 departments, approximately 35% of the Ph.D.'s are awarded to domestic students.

Methodology of Admissions Study

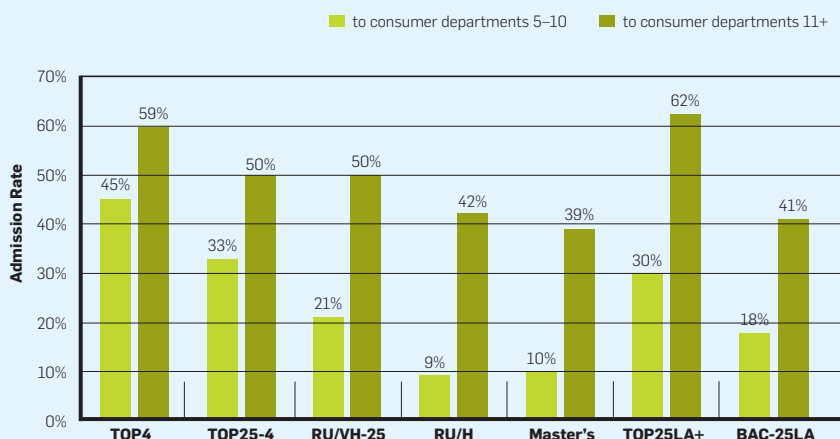
Our admissions study partitioned the 14 departments providing admission records into those ranked 5–10 and those ranked 11+. We refer to these 14 departments as the *consumers* since they take students into their graduate programs. The institutions where the 7,032 students completed their undergraduate studies are called the *producers*. Approximately half of the 7,032 graduate records came from consumer departments ranked 5–10 and half from departments ranked 11+. We intentionally did not include admissions data to consumer departments ranked 1–4 since their graduate admissions profiles are significantly different (their acceptance rates are typically one-third of those for schools ranked 5–10).

Admission records are for domestic students having completed a baccalaureate degree at a producer institution. Each record includes the following information: the name of the producer school, year the student applied to graduate school, name of the consumer school the student applied to, decision of whether or not the student was accepted by the consumer school, and—if the student was accepted—whether or not the student chose to matriculate. The data also included gender, underrepresented minority status, GPA, and GRE scores.

The dataset has some inherent limitations. Admission records had no names and thus students who applied to multiple graduate programs produced repeated records. Records do not include important information used in the admission process, such as recommendation letters, statements of purpose, and research achievements. Finally, there is no way of knowing which applicants accepting admission actually did/will receive a Ph.D.

The 7,032 records came from a wide range of producer schools. To better identify and understand trends, we

Figure 1. Admission rates of applications from the seven producer groups to consumer departments ranked 5–10 and 11+.



partitioned the producer schools into seven groups (listed in the accompanying table). We use a commonly used classification of institutions, the 2005 Basic Classification of the Carnegie Foundation,⁵ and the most frequently used rankings of CS departments and liberal arts colleges, the *USNWR* rankings.⁶ These seven groups capture over 91% of all applications.

Findings

For the entire dataset, 35% of the applications resulted in admission. Figure 1 shows the admission rates from each of the seven producer groups (x-axis) to the participating consumers (graduate schools) ranked 5–10 and ranked 11+. While it is not surprising that schools ranked 5–10 have lower admission rates than schools ranked 11+, the disparity is particularly notable for producer groups RU/H and master's. For example, for students receiving their baccalaureate degree from an RU/H institution, admission rates are 9% for applications to departments ranked 5–10 versus 42% for schools ranked 11+.

Figure 2 summarizes the number of applications, admissions, and matriculations to consumer departments ranked 5–10. Figure 3 does the same for consumers departments ranked 11+. For example, 927 of the 7,032 records are from students with baccalaureate degrees from RU/VH-25 departments applying to graduate programs ranked 5–10. From those, 196 were accepted (21% acceptance rate) and 104 matriculated (53% matriculation rate).

Figure 2 shows that consumer departments ranked 5–10 have a matriculation rate of close to 50% or better for all producer groups, except for students from TOP25LA+ producers schools. Their matriculation rate is only about 28%. For consumer departments 11+, the lowest admission rate is observed for students from the master's producer group (39%). Admission rates are about 60% for students from TOP4 and TOP25LA producer groups. Matriculation rates range from about 20% (for students from the TOP4 and TOP25LA producer groups) to about 40%.

For 99% of the admission records the data included gender, and for 76% it included ethnicity. The percentage

of female applicants was 14%, and the percentage of African Americans and Hispanic/Latino was 3% each. Comparing this to their percentage of Ph.D.'s awarded, approximately 17% of CS Ph.D. graduates in the U.S. are women, 1.5% are Hispanic, and 1.5% are African American.¹

Our dataset showed some interesting trends with respect to gender and ethnicity. In particular, a disproportionate fraction of female applicants came from TOP25 liberal arts colleges, while a disproportionately small fraction of African American and Hispanic students came from these colleges. A relatively large fraction of Hispanic applications came from RU/VH-25 producer institutions, while RU/H and BAC-25LA were major producers of African American applicants.

Observations and Recommendations

Looking at the baccalaureate origins of students in the domestic pipeline, a number of trends emerge. Master's institutions and departments awarding Ph.D.'s produce approximately an equal number of computer science bachelor's degrees.³ But the number of applications accepted for admission at Ph.D. programs ranked 5–10 from students who received their bachelor's from research universities was over 24 times higher than for students from master's institutions. The analogous ratio for the 11+ consumer departments was 4.5.

Members of graduate admission committees are, understandably, cautious about admitting students who do not appear prepared for graduate

Figure 2. Number of applications, admissions, and matriculations to consumer departments ranked 5–10.

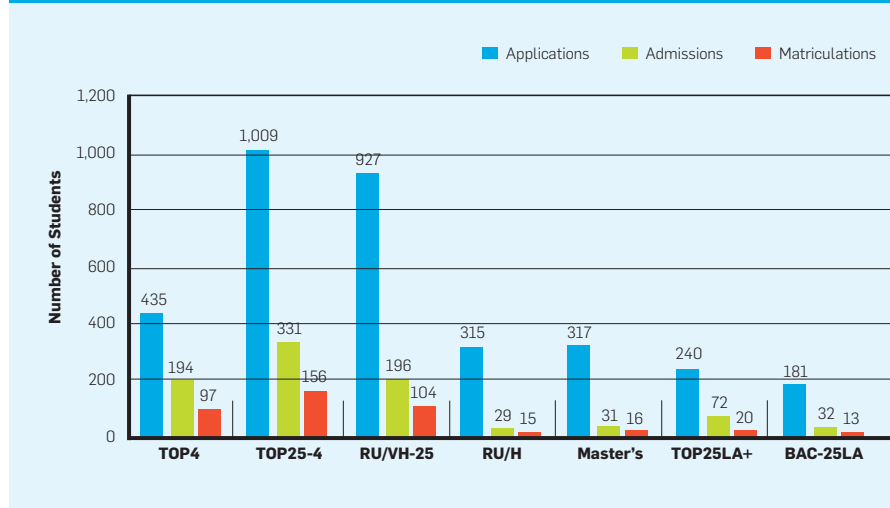
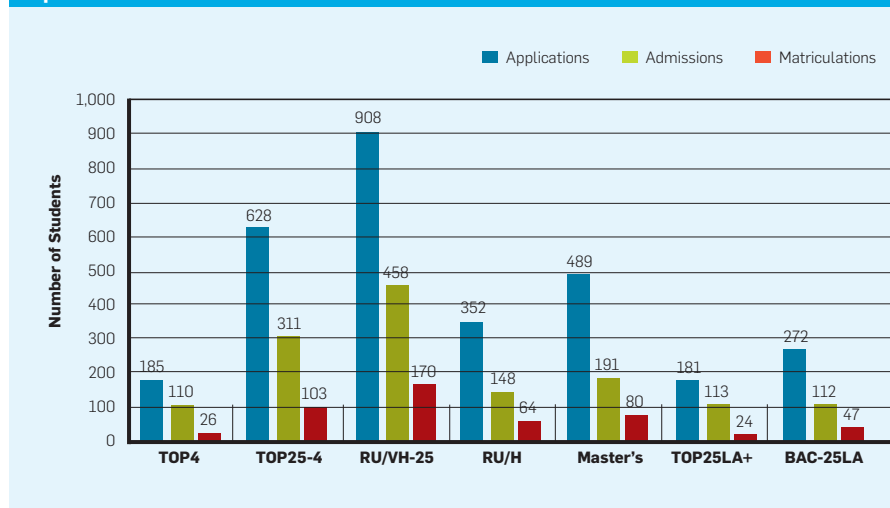


Figure 3. Number of applications, admissions, and matriculations to consumer departments ranked 11+.



COMMUNICATIONS APPS



Association for
Computing Machinery

Access the
latest issue,
past issues,
BLOG@CACM,
News, and
more.



Available for iPad,
iPhone, and Android



Available for iOS,
Android, and Windows

Graduate admissions committees need to be realistic about their applicant pools.

work, particularly if they have had no firsthand research experience. But talented students with potential for research exist in all producer groups. Consumer departments interested in increasing the number of domestic Ph.D. students, and producer schools wanting to better advise their students to be competitive for graduate school, should consider the following interventions and strategies.

Graduate admissions committees need to be realistic about their applicant pools. Applicants from TOP25 departments are highly desirable, and active recruiting from TOP25 departments may not be as effective as expected. Even for highly ranked Ph.D. programs, matriculation rates for these students are low.

Graduate admissions committees should also be aware of the breadth and differences of educational institutions in the U.S. Our findings indicate some large graduate schools have forged productive partnerships with colleges and smaller universities in their regions. These relationships can begin with a faculty member from the Ph.D. program giving a talk to undergraduates and faculty at prospective partner institutions, and it can evolve into summer and academic-year research opportunities for undergraduate students from the partner colleges and universities.

Producer departments should support and foster computer science student research communities and highlight student research achievements. One effective practice is to hold annual undergraduate research presentations and poster sessions and provide incentives (such as food and beverages) to attract younger undergraduates to see what their peers have done.

Publicized recognition of undergraduate research achievements can also be effective.

Producer departments should hold periodic information sessions on careers in research, summer research experiences, and graduate school, and should invite researchers from industry and academia to speak with students about careers in research. Undergraduates—even those at research universities—often have misconceptions about research and graduate school.

Private and federal research laboratories can contribute to the health of the domestic pipeline by expanding the quantity and breadth of undergraduate research opportunities and encouraging summer interns to pursue graduate studies.

For all producer and consumer institutions, the Computer Science Undergraduate Research CONQUER website² has valuable information and resources for supporting undergraduate involvement in research. This information can be useful in any student research environment, and in particular, it can be a resource to bring more students into the domestic Ph.D. pipeline. **C**

References

1. CRA Taulbee Survey; <http://cra.org/resources/taulbee/>.
2. CRA-E's Undergraduate Research Conquer website (CONQUER); <http://cra.org/conquer>.
3. Hambrusch, S. et al. Exploring the baccalaureate origin of domestic Ph.D. students in computing fields. *Computing Research News* (Jan. 2013).
4. Hambrusch, S. et al. Findings from a pipeline study based on graduate admissions records. In *Proceedings of the CRA Snowbird Conference 2014*; <http://cra.org/crae/reports>.
5. The Carnegie Classification of Institutions of Higher Education; <http://classifications.carnegiefoundation.org/>.
6. *U.S. News and World Report*. Computer science rankings (2014); <http://grad-schools.usnews.rankingsandreviews.com/best-graduate-schools/top-science-schools/computer-science-rankings>.

Susanne Hambrusch (seh@cs.purdue.edu) is a professor of computer sciences at Purdue University in West Lafayette, IN.

Ran Libeskind-Hadas (hadas@cs.hmc.edu) is the R. Michael Shanahan Professor and Department Chair of the Department of Computer Science at Harvey Mudd College in Claremont, CA.

Eric Aaron (eraaron@vassar.edu) is a visiting assistant professor of computer science at Vassar College in Poughkeepsie, NY.

The authors are members of the Computing Research Association Education Committee and gratefully acknowledge valuable contributions by Charles Isbell and Elijah Cameron of the College of Computing at the Georgia Institute of Technology.

Copyright held by authors.

Viewpoint

Learning Through Computational Creativity

Improving learning and achievement in introductory computer science by incorporating creative thinking into the curriculum.

THE NATIONAL SCIENCE Foundation's "Rebuilding the Mosaic" report^a notes that addressing emerging issues in all fields will require utilization and management of large-scale databases, creativity in devising data-centric solutions to problems, and application of computational and computer tools through interdisciplinary efforts. In response to these needs, introductory computer science (CS) courses are becoming more than just a course for CS majors. They are becoming multipurpose: designed not just to prepare future CS scientists and practitioners, but also to inspire, motivate, and recruit new students to CS; provide computational thinking tools and CS skills to students in other disciplines; and even train future CS K-12 teachers. This multifaceted nature of introductory CS calls for new ways of envisioning the CS curriculum. Along with computational thinking, creativity has been singled out as critical to addressing important societal problems and central to 21st century skills (for example, the 2012 National Research Council report). Driven by these observations, we see an opportunity to revise introductory CS courses by explicitly integrating creative thinking into the curriculum.

Creative Thinking

Creative thinking is not an innate talent or the province of just a few individ-



uals, and it is not confined to the arts. Rather, it is a process integral to human intelligence that can be practiced, encouraged, and developed within any context.^{1,2,5,7-9} Epstein's Generativity Theory² breaks creative thinking down to four core competencies:

► **Broadening.** The more diverse one's

knowledge and skills, the more varied and interesting the possible novel patterns and combinations that might emerge. To be creative one must broaden one's knowledge by acquiring information and skills outside one's current domains of study and expertise.

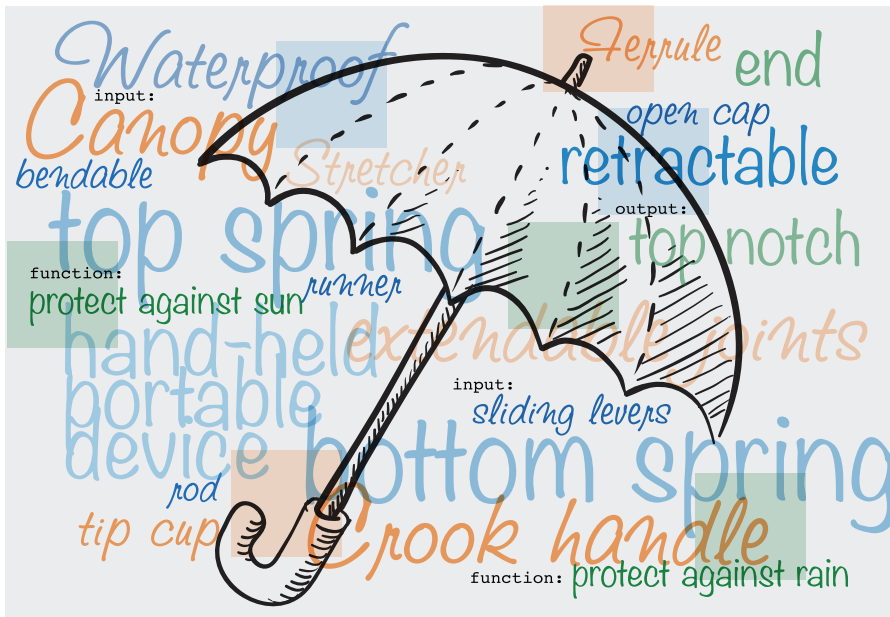
► **Challenging.** Novelty emerges from situations where existing strategies and behaviors are ineffective. The more difficult the challenge, the more likely a creative, novel solution will emerge.

► **Surrounding.** Exposure to multiple, ambiguous situations and stimuli create environments where novel strategies and behaviors may emerge—for example, looking at things in new ways, interacting with new people, and considering multiple sensory representations.

► **Capturing.** Novelty is occurring all the time, but most of it passes without recognition. Creativity requires attention to and recording of novelty as it occurs.

These core competencies have a solid anchoring in contemporary cognitive and neuroscience research.⁶ Just as Wing^{10,11} makes a convincing case for the universality of computational thinking, we argue that Epstein's core creative thinking competencies are also a universally applicable skill set that provides a foundation not only for applying one's existing knowledge and skills in creative ways, but also for engaging in lifelong learning to broaden one's capabilities for work in interdisci-

a See <http://www.nsf.gov/pubs/2011/nsf11086/nsf11086.pdf>.



plinary environments on increasingly complex problems.

Computational Creativity Exercises

In our framework, both computational thinking and creative thinking are viewed as cognitive tools that when blended form *computational creativity*. This blending is not conceived as a dichotomy, but rather as symbiotic abilities and approaches. Computational thinking and CS skills expand the knowledge and tools that one has available, thereby *broadening* the scope of problem solutions. *Challenging* problems force computational tools to be used in unanticipated and unusual ways, leading to new computational approaches to both old and new problems. *Surrounding* oneself with new environments and collaborators creates novel ways of looking at problems and attention to different stimuli and perspectives that may be relevant to approaching a problem computationally. Finally, *capturing* ideas for data representations and algorithms can lead to new data structures and solution procedures. By merging computational and creative thinking, students can leverage their creative thinking skills to “unlock” their understanding of computational thinking.⁶

We have created a suite of Computational Creativity Exercises (CCEs) designed to increase students’ computational creativity. Each CCE has four common components: Objectives, Tasks, CS Lightbulbs—explanations

connecting activities to CS concepts, ideas, and practices—and Learning Objects that relate the exercise tasks directly to CS topics. The principles underlying the design of our Computational Creativity Exercises are balancing of attributes between computational and creative thinking and mapping between computational and creative concepts and skills as they manifest in different disciplines. Each CCE requires approximately one to two hours per student, but students are given two weeks to work on the exercises because of the collaboration required. The CCEs are designed so that the students have hands-on and group tasks first, in Week 1, and then reflect on their Week 1 activities in Week 2 by answering analysis and reflection questions. Both Week 1 and Week 2 are graded.

We see an opportunity to revise introductory CS courses by explicitly integrating creative thinking into the curriculum.

As an example, in the *Everyday Object CCE*, students are asked to act like the inventor of an ordinary object that we might frequently use. The challenge is to imagine this object does not exist and to describe in written language: the mechanical function of the selected object; what need is fulfilled by this object; and its physical attributes and characteristics. The description must be specific enough so that someone who had never seen the object could recognize it and understand how it works and understand what benefits it provides. (Note: Students were given a list of objects—zipper, mechanical pencil, binder clip, Ziploc bag, scissors, tape measure, stapler, nail clippers, umbrella, flashlight, can opener, clothespin, sticky notes, toilet paper holder, revolving door—from which to choose.) Students are then asked to consider and write their responses to the following questions. **Analysis:** (1) Consider your object as a computer program. Draw a diagram that shows all its functions as boxes (name them), and for each function, its inputs and outputs. Are there shared inputs and outputs among the functions? (2) Consider the list of physical attributes and characteristics. Organize these such that each is declared as a variable with its proper type. Can some of these attributes/characteristics be arranged into a hierarchy of related attributes/characteristics? **Reflection:** (1) Consider your response to Analysis 1, are there functions that can be combined so that the object can be represented with a more concise program? Are there new functions that should be introduced to better describe your object such that the functions are more modular? (2) Have you heard of abstraction? How does abstraction in computer science relate to the process of identifying the functions and characteristics as you have done in this exercise?

Our CCEs are anchored in instructional design principles shown to impact deep learning, transfer, and development of interpersonal skills. They are designed to provide instruction on CS concepts by combining hands-on problem-based learning with written analysis and reflection. They facilitate transfer by using computational thinking and CS content more abstractly and

without using programming code to address problems seemingly unrelated to CS. The CCEs foster development of creative competencies by engaging multiple senses, requiring integrative, imaginative thought, presenting challenging problems and developing interpersonal skills using individual and collaborative group efforts. The CCEs engage the cognitive/neural learning processes of *attention*, *repetition*, and *connection* identified in the Unified Learning Model⁶ on synthesis of research in cognitive neuroscience, cognitive science, and psychology. They enhance learning and retention of course material by focusing attention on computational thinking principles, provide additional repetition of computational thinking and computing concepts from the class, and provide connections of the material to more diverse contexts and applications at a higher level of abstraction.

Some Evidence

Four CCEs were deployed in four different introductory computer science courses during the Fall 2012 semester at the University of Nebraska, Lincoln. Each course was tailored to a different target group (CS majors, engineering majors, combined CS/physical sciences majors, and humanities majors). Findings from 150 students showed that with cumulative GPA controlled, the number of CCEs completed was significantly associated with course grade ($F(3, 109) = 4.32, p = .006$, partial $\eta^2 = .106$). There was a significant linear trend ($p = .0001$) from 2 to 4 exercises completed. The number of CCEs completed also was significantly associated with a computational thinking knowledge test score ($F(3, 98) = 4.76, p = .004$, partial $\eta^2 = .127$), again with a significant linear trend ($p < .0001$) from 0–1 to 4 exercises completed with no differences for CS and non-CS majors.^{3,4} These findings indicated a “dosage” effect with course grades and test scores increasing with each additional CCE completed. The increases were *not* trivial. Students increased by almost a letter grade and almost one point on the knowledge test per CCE completed.

In a second evaluation,⁷ we employed a quasi-experimental design comparing CCE implementation in the introductory computer science

These core competencies have a solid anchoring in contemporary cognitive and neuroscience research.

course tailored for engineers during Spring 2013 ($N = 90$, 96% completing three or four exercises) to a control semester ($N = 65$) of no implementation in Fall 2013. Using Analysis of Covariance (ANCOVA), we found that students in the implementation semester had significantly higher computational thinking knowledge test scores than students in the control semester ($M = 7.47$ to $M = 5.94$) when controlling for students' course grades, strategic self-regulation, engagement, motivation, and classroom perceptions. ($F(1, 106) = 12.78, p < .01$, partial $\eta^2 = .108$). CCE implementation students also reported significantly higher self-efficacy for applying their computer science knowledge and skills in engineering than non-implementation students ($M = 70.64$ to $M = 61.47$; $F(1, 153) = 12.22, p < .01$, partial $\eta^2 = .074$).

Overall, in relation to traditional findings for classroom educational interventions, these are strong effects that demonstrate meaningful “real-world” impact. The exercises appear to positively affect the learning of core course content and achievement for both CS majors and non-majors. The findings support our contention that *the Computational Creativity Exercises can bring CS computational concepts to CS and non-CS disciplines alike and improve students' understanding of computational thinking.*

Call to Action

Encouraged by our evaluation findings, we are currently working on adapting the CCEs to secondary schools, designing a course based on a suite of CCEs, as well as continuing with the develop-

ment of more CCEs. Furthermore, the broader impacts of incorporating computational creativity into introductory CS courses are wide-ranging including reaching underrepresented groups in CS, outreach exposing young learners to computational creativity, improving learning of CS, and preparing students to be creative problem solvers in increasingly interdisciplinary domains. We thus call to action CS (and other STEM) educators to investigate and adopt computational creativity in their courses or curricula. **C**

References

1. Epstein, R. *Cognition, Creativity, and Behavior: Selected Essays*. Praeger, 1996.
2. Epstein, R. Generativity theory and creativity. *Theories of Creativity*. Hampton Press, 2005.
3. Miller, L.D. et al. Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses. *FIE* 43, (2013), 1426–1432.
4. Miller, L.D. et al. Integrating computational and creative thinking to improve learning and performance in CS1. *SIGCSE'2014* (2014), 475–480.
5. Robinson, K. *Out of Our Minds: Learning to be Creative*. Capstone, 2001.
6. Shell, D.F., Brooks, D.W., Trainin, G., Wilson, K., Kauffman, D.F., and Herr, L. *The Unified Learning Model: How Motivational, Cognitive, and Neurobiological Sciences Inform Best Teaching Practices*. Springer, 2010.
7. Shell, D.F. et al. Improving learning of computational thinking using computational creativity exercises in a college CS1 computer science course for engineers. *FIE* 44, to appear.
8. Shell, D.F. and Soh, L.-K. Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses. *J. Sci. Edu. Tech. Technology* (2013).
9. Tharp, T. *The Creative Habit: Learn it and Use it for Life*. Simon & Schuster, 2005.
10. Wing, J. Computational thinking. *Commun. ACM* 49, 3 (Mar. 2006), 33–35.
11. Wing, J. Computational thinking: What and why. *Link Magazine* (2010).

Leen-Kiat Soh (lksoh@cse.unl.edu) is an associate professor in the Computer Science and Engineering Department at the University of Nebraska.

Duane F. Shell (dshell2@unl.edu) is a research professor at the University of Nebraska.

Elizabeth Ingraham (eingraham2@unl.edu) is an associate professor of art at the University of Nebraska.

Stephen Ramsay (sramsay.unl@gmail.com) is the Susan J. Rosowski Associate Professor of English at the University of Nebraska.

Brian Moore (brian.moore@unl.edu) is associate professor of music education and music technology at the University of Nebraska.

This material is based upon work supported by the National Science Foundation under grant no. 1122956. Additional support was provided by a University of Nebraska-Lincoln (UNL) Phase II Pathways to Interdisciplinary Research Centers grant. Any opinions, findings, conclusions, or recommendations expressed in our materials are solely those of the authors and do not necessarily reflect the views of the National Science Foundation or UNL.

Article development led by **acmqueue**
queue.acm.org

Use states to drive your tests.

BY ARIE VAN DEURSEN

Testing Web Applications with State Objects

END-TO-END TESTING OF Web applications typically involves tricky interactions with Web pages by means of a framework such as Selenium WebDriver.¹² The recommended method for hiding such Web-page intricacies is to use page objects,¹⁰ but there are questions to answer first: Which page objects should you create when testing Web applications? What actions should you include in a page object? Which test scenarios should you specify, given your page objects?

While working with page objects during the past few months to test an AngularJS (<https://angularjs.org>) Web application, I answered these questions by moving page objects to the *state* level. Viewing the Web application as a state chart made it much easier to design test scenarios and corresponding page objects. This article describes the approach

that gradually emerged: essentially a state-based generalization of page objects, referred to here as *state objects*.

WebDriver is a state-of-the-art tool widely used for testing Web applications. It provides an API to access elements on a Web page as they are rendered in a browser. The elements can be inspected, accessing text contained in tables or the element's styling attributes, for example. Furthermore, the API can be used to interact with the page—for example, to click on links or buttons or to enter text in input forms. Thus, WebDriver can be used to script click scenarios through a Web application, resulting in an end-to-end test suite of an application.

WebDriver can be used to test against your browser of choice (Internet Explorer, Firefox, Chrome, among others). It comes with different language bindings, allowing you to program scenarios in C#, Java, JavaScript, or Python.

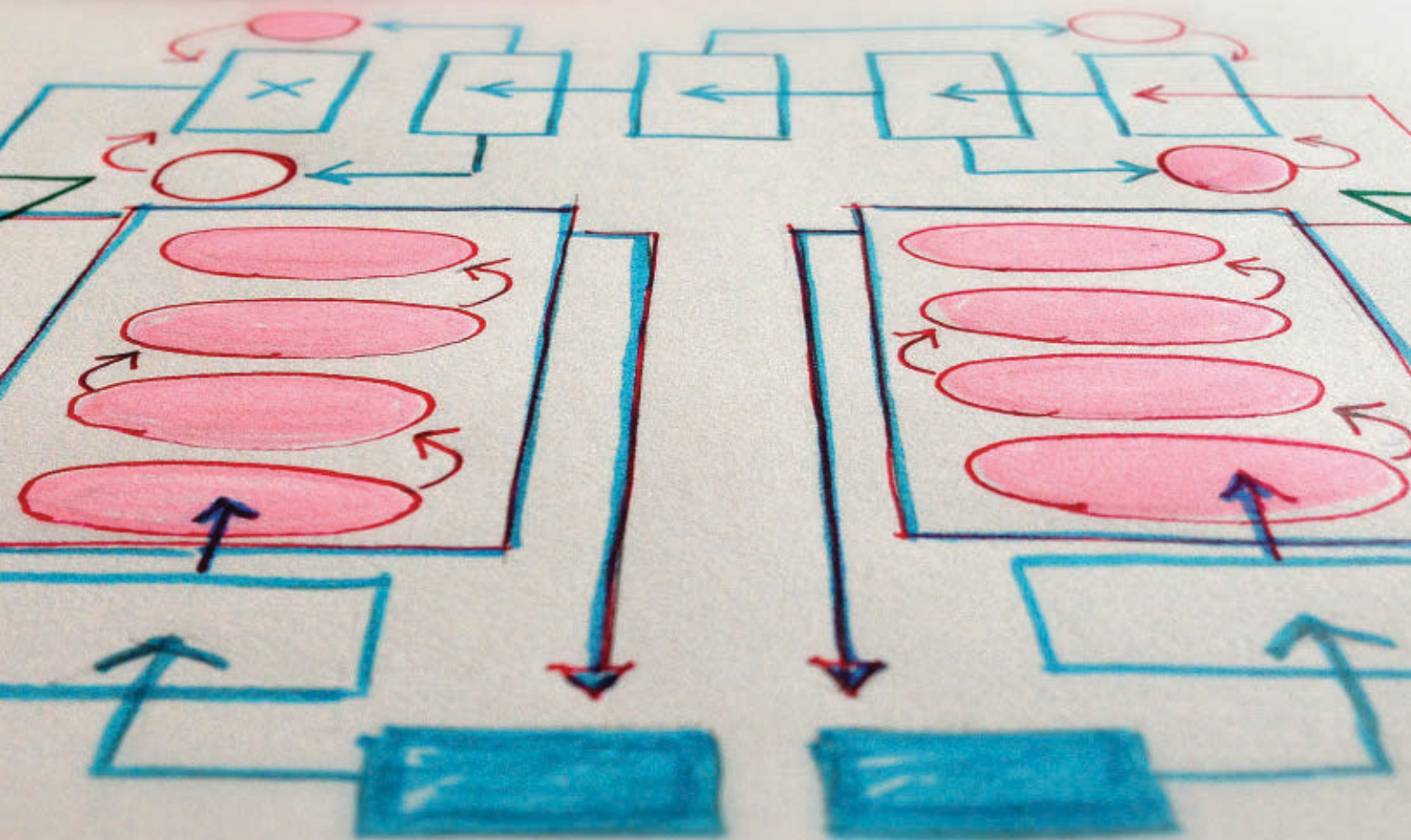
Page Objects. WebDriver provides an API to interact with elements on a Web page as rendered in a browser. Meaningful end-user scenarios, however, should not be expressed in terms of Web elements, but in terms of the *application domain*.

Therefore, a recommend pattern for writing WebDriver tests is to use page objects. Such objects provide an API over domain concepts implemented on top of Web-page elements, as illustrated in Figure 1, taken from an illustration by software designer Martin Fowler.⁴

These page objects hide the specific element locators used (for example, to find buttons or links) and the details of the underlying “widgetry.” As a result, the scenarios are more readable and easier to maintain if page details change.

Page objects need not represent a full page; they can also cover part of the user interface such as a navigation pane or an upload button.

To represent navigation through the application, “methods on the Page-Object should return other PageOb-



jects.”¹⁰ It is this idea that this article takes a step further, leading to what we will refer to as *state objects*.

Modeling Web Apps with State Charts

To model Web application navigation, let’s use statecharts from the Unified Modeling Language (UML). Figure 2 shows a statechart used for logging into an application. Users are either authenticating or authenticated. They start not being authenticated, enter their credentials, and if they are OK, they reach the authenticated state. From there, they can log off to return to the page where they can authenticate.

This diagram traditionally leads to two page objects:

- One for the login page, corresponding to the authenticating state.

- One for the logoff button, present on any page shown in the authenticated state.

To emphasize these page objects represent states, they are given explicit responsibilities for state navigation and state inspection, and they become state objects.

State Objects: Checks and Trigger Methods

Two types of methods can be identified for each state object:

- *Inspection methods* return the value of key elements displayed in the browser when it is in the given state, such as a user name, the name of a document, or some metric value; They can be used in test scenarios to verify the browser displays the expected values.

- *Trigger methods* correspond to an imitated user click and bring the

browser to a new state. In the authenticating state users can enter credentials and click the submit button, which, assuming the credentials are correct, leads the browser to the next authenticated state. From there, the user can click the logoff button to get back to the authenticating state.

It is useful to combine the most important inspection methods into one self-check of properties that must hold whenever the application is in that particular state. For example, on the authenticating state, you would expect fields for entering a user name or a password; there should be a submit button; and perhaps the URL should include the login route. Such a self-check method can then be used to verify the browser is indeed in a given state.

Scenarios: Triggering and checking events. Given a set of state objects,

Figure 1. Page objects. (Figure by Martin Fowler)

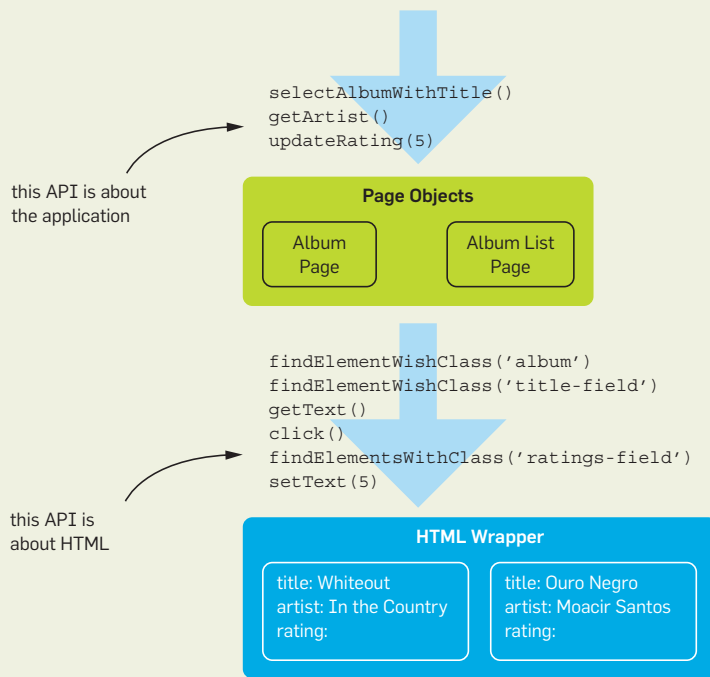
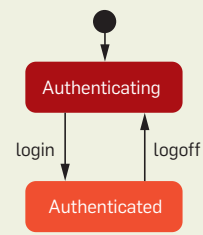


Figure 2. Logging into an application.



1. Go to the login URL.
2. Conduct the Authenticating self-check.
3. Enter invalid credentials and submit.
4. Conduct the Login Error self-check.
5. Hit close.
6. Conduct the Authenticating self-check.

In Figure 3 the edges are of the form

event [condition] / action

Thus, a trigger (click) can be conditional, and besides leading to a new state it can also result in an action (server side).

When testing such transitions, you trigger the event, ensure the condition is met, and then verify: that you can observe any effects of the required action; and that you reach the corresponding state.

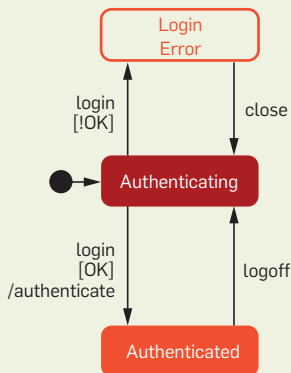
Expanding Your State Chart

To drive the testing, you can expand the state chart to cater to additional scenarios. For example, related to authentication is registering a new user, shown in Figure 4. This Figure includes the Authenticating state, but not all of its outgoing edges. Instead, the focus is on a new *Registering* state and the transitions that are possible from there.

This again gives rise to two new state objects (for registration and for displaying an error message), and two additional scenarios. Thus, when developing tests for a given page, it is not necessary to consider the full state machine: focusing on states of interests is sufficiently helpful for deriving test cases.

Super states. States that have behavior in common can be organized into *super states* (also called OR-states). For example, once authenticated, all pages may have a common header, containing buttons for logging out, as well as

Figure 3. Login with conditional events.

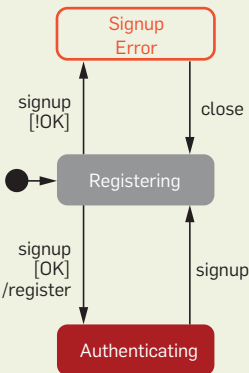


test cases describe relevant *scenarios* (paths) through the state machine. For example:

1. Go to the login URL.
2. Verify you are in the Authenticating state via self-check.
3. Enter correct credentials and submit.
4. Verify you are in the Authenticated state.
5. Hit logoff.
6. Verify you are in the Authenticating state.

Thus, a scenario (test or acceptance) is a sequence of actions, each followed by a check the expected state

Figure 4. Registering a new user.



has been reached.

Conditional events. Besides successful login, a realistic login procedure should also handle attempts to log in with invalid credentials, as shown in Figure 3. The figure shows an extra state in which an error message is displayed. This extra state gives rise to an extra state object, corresponding to the display of the appropriate message. As action, it just has a close button leading back to the original login page.

The extra state naturally leads to another test scenario:

for navigating to key pages (for example, for managing your account or obtaining help, shown in Figure 5).

Edges going out of the super state (such as `logout`) are shorthand for an outgoing `logout` event for each of the four internal states (the substates). Expanding the shorthand would lead to the diagram in Figure 6 where the five outgoing edges of the super state are expanded for each of the four internal states, leading to $4 * 5 = 20$ (directed) edges (drawn as two-way edges to keep the diagram manageable).

The typical way of implementing such super states is by having reusable HTML fragments, which in AngularJS, for example, are included via the `ngInclude` directive.¹

In such cases, it is most natural to create a state object corresponding to the common `include` file. It contains presence checks for the required links or buttons and event checks to see if, for example, clicking settings indeed leads to the Settings state.

A possible test scenario would then be:

1. [Steps needed for login.]
2. Conduct Portfolio self-check.
3. Click settings link.
4. Conduct Settings self-check.
5. Click help link.
6. Conduct Help self-check.
7. Click account link.
8. Conduct Account self-check.
9. Click portfolio link.
10. Conduct Portfolio self-check.
11. Click logout link.
12. Conduct Authenticating self-check.

This corresponds to a single scenario testing the authenticated navigation pane. It tests that clicking the account link from the Help page works. It does not, however, check that clicking the account link from the Settings page works. In fact, this test covers only four of the 20 edges in the expanded graph.

Of course you can create tests for all 20 edges. This may make sense if the app-under-test has handcrafted the navigation pane for every view instead of using a single `include` file. In that case you may have reason to believe the different trails could reveal different bugs. Usually, however, testing all expanded paths would be overkill for the `include` file setting.

State traversals. The single test scenario for the navigation header visits

five different states, in one particular order, shown in Figure 7. This is a rather long test and could be split into four separate test cases (Figure 8).

In unit testing, the latter would be the preferred way. It has the advantage of the four tests being independent: failure in one of the steps does not affect testing of the other steps. Moreover, fault diagnosis is easier, since the failed test case will be clearer.

The four independent tests, however, are likely to take considerably longer: every test requires explicit authentication, which will substantially slow down test execution. Therefore, in end-to-end testing it is more common to see shared setup among test cases.

In terms of JUnit's (<http://junit.org>) setup methods,⁹ a unit test suite would typically make use of the `@Before` set-

Figure 5. Common behavior in a super state.

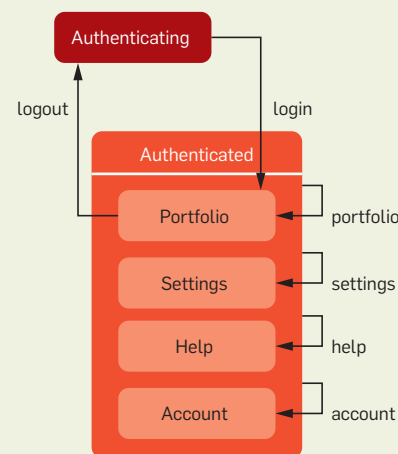


Figure 7. Single test traversing multiple states.

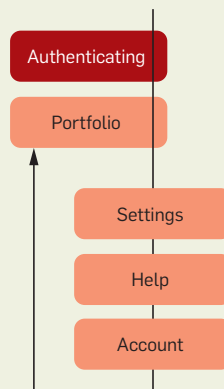


Figure 6. Expanding a super state.

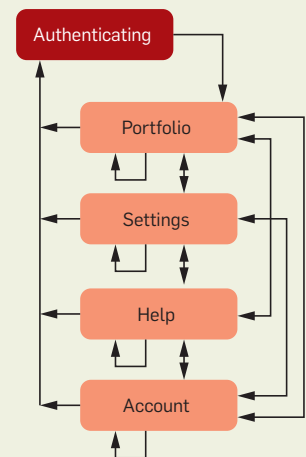


Figure 8. Separate tests for different states.

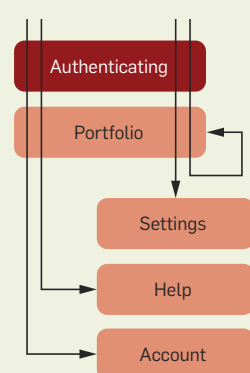


Figure 9. Login state machine with error handling and shared navigation.

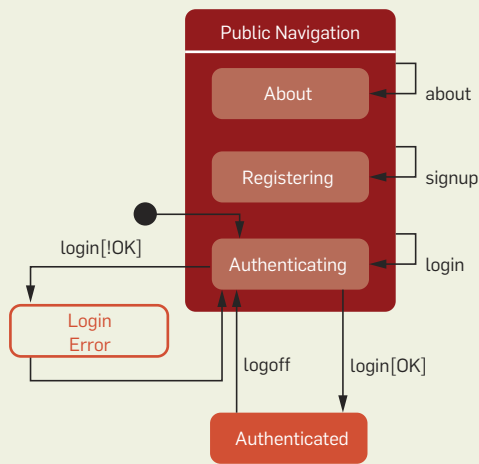
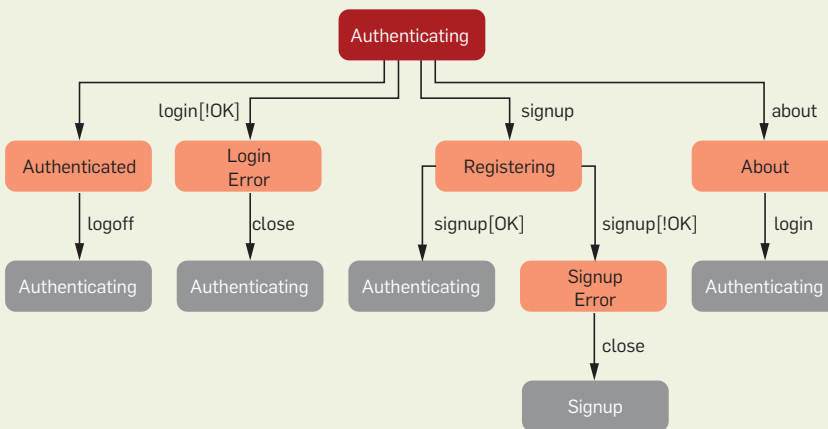


Figure 11. Transition tree.



(that is, that any other interaction with the page is disabled).

If the modal dialog is triggered by a state contained in a super state, the dialog state is *not* part of the super state (since the super-state interaction is disabled in the dialog). Thus, the correct way to draw the login state machine showing error handling and shared navigation would be as illustrated in Figure 9. Here the error dialog is not part of the navigation super state, as it permits only the close event and not clicking, for example, the about link.

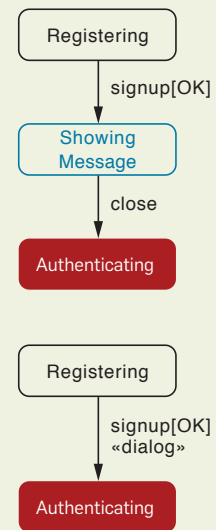
Some applications are fairly dialog-intensive—for example, when dealing with registration, logging in, forgetting passwords, among others. Many

of these dialogs serve only to notify the user of a successful state change. To simplify the state diagrams the dialogs can then be drawn as annotations on the edges, as in Figure 10.

The diagram at the top is the full version, and the one at the bottom is the abbreviated version. Note the `<<dialog>>` annotation is important for implementing the test. The test cases *must* click the close button of the dialog; otherwise, testing is blocked.

The transition tree. To support reasoning about state reachability, as well as state and transition coverage, it is helpful to turn a state diagram into a transition tree, as shown in Figure 11. (For more information, see Robert Binder's chapter on test-

Figure 10. Using transition annotations to simplify diagrams.



ing state machines in *Testing Object-oriented Systems*.³⁾

The tree in Figure 11 has been derived from the state machine showing sign-up and authentication as presented earlier. Starting from the initial authenticating state, let's do a breadth-first traversal of the graph. Thus, per state you first visit its direct successors. If you enter a state you visited already, the visited state is drawn in gray as a leaf node. Then proceed to the next unvisited state.

The tree helps when designing tests for an individual state: the path to that state in the tree is the shortest path in the graph to that state. Besides, it clearly indicates which outgoing edges there are for a state.

The tree is also helpful for designing a test suite for the full state machine: writing one test case for each path from the root to a leaf yields a test suite that covers all transitions and, hence, covers all states in the machine.

Covering paths. While focusing on individual states and their transitions is a good way to spot and eliminate basic faults, a trickier set of defects is visible only when following a *path* along multiple states.

As an example, consider client-side caching. A framework such as AngularJS makes it easy to enable caching for (some) back-end HTTP calls. Doing this right improves responsiveness and reduces network round-trips, since the

results of back-end calls are remembered instead of requested over and over again.

If, however, the results are subject to change, this may lead to incorrect results. For example, the client may request an overview page with required information on one page, modify the underlying data on the next page, and then return to the original overview page. This corresponds to the red path in Figure 12.

With caching enabled, the Portfolio state will cache the back-end call results. The correct caching implementation of the Settings state would be to invalidate the cache if changes were made. As a result, when revisiting the Portfolio state the call will be made again, and the updated results will be used.

A test case for this caching behavior might look as follows:

1. [Take shortest route to Portfolio.]
2. Collect values of interest from Portfolio.
3. Click the settings link to navigate to Settings.
4. Modify settings that will affect Portfolio values of interest.
5. Click the portfolio link to navigate back to Portfolio.
6. Assert that modified values are displayed.

In the AngularJS application mentioned previously, this test case caught an actual bug. Unfortunately, it is difficult or expensive to come up with a test strategy that covers *all* such paths possibly containing bugs.

In the general case, in the presence of loops there are infinitely many potential paths to follow. Thus, the tester will need to rely on expertise to identify paths of interest.

The transition tree-based approach described previously provides so-called round-trip coverage²—that is, it exercises each loop once until it gets back at a node already visited (one round-trip). Assuming all super states are expanded, this strategy would lead to a test case for the caching example.

Alternative criteria include all length-N paths, in which every possible path of a given length must be exercised. The extra costs in terms of the increased number of test cases to be written can be substantial, however, so achieving such a criterion without automated tools is typically hard.

In terms of state objects, testing paths will not lead to new state objects—the states are already there. The need to assert properties along the path, however, may call for additional inspection methods in the state objects.

Going Backward

The browser's back button provides state navigation that requires special attention. While this button makes sense in traditional hypertext navigation, in today's Web applications it is not always clear what its behavior should be.

Web developers can alter the button's behavior by manipulating the history stack. As such, you want to be able to test a Web application's back-button behavior, and WebDriver provides an API call for it.

In terms of state machines, the back button is not a separate transition. Instead, it is a reaction to an earlier (click) event. As such, back-button behavior is a property of an edge, indicating the transition can be “undone” by following it in the reverse direction.

UML's mechanism for giving special meaning to elements is to use annotations (profiles). In Figure 13 explicit `<<back>>` and `<<noback>>` annotations have been added to the edges to indicate whether the back button can be used after the click to return to the initiating state. Thus, for simple navigation between the About, Registering, and Authenticating states, the back button can be used to navigate back.

Between the Authenticated and Authenticating state, however, the back button is effectively disabled: once logged off, clicking “Back” should not allow anyone to go to content requiring authentication. Knowing which transitions have special back behavior will then guide the construction of extra test scenarios verifying the required behavior.

Super States with History

As a slightly more sophisticated example of a super state, consider a table that is sortable across different columns. Clicking on a column header causes the table to be sorted, giving rise to a substate for every column (Figure 14).

The events come out of the super state, indicating they can be triggered from any substate and go into a par-

Figure 12. Faults on longer paths.

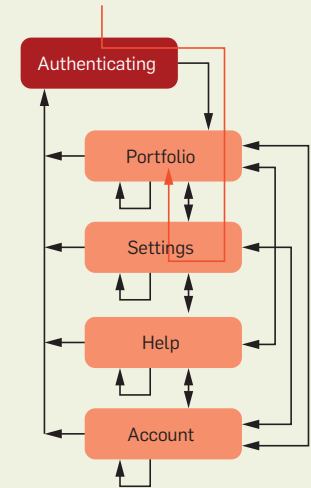


Figure 13. Back-button annotations.

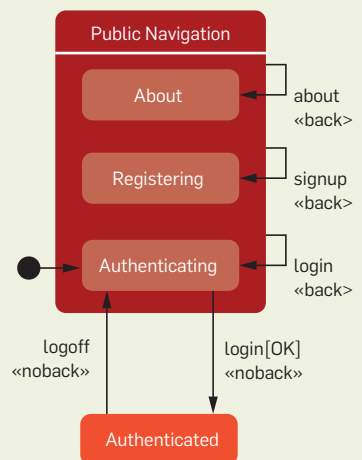


Figure 14. Super state with history.

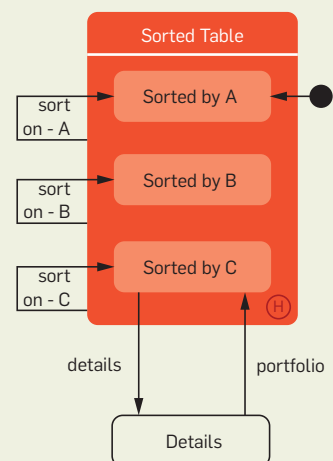


Figure 15. Orthogonal regions.

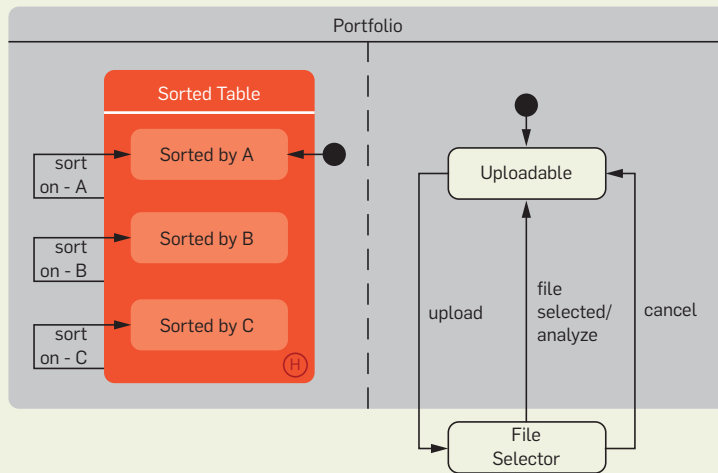
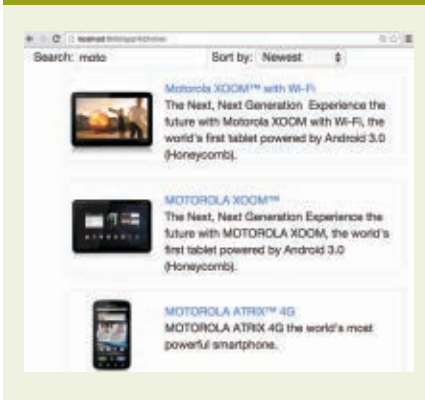


Figure 16. PhoneCat AngularJS Application.

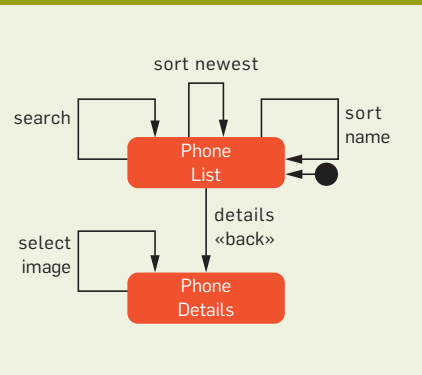


ticular substate. When leaving the sortable table page—for example, by requesting details for a given row—a design decision needs to be made about whether returning to that page (in this case by clicking the portfolio link) should yield a table sorted by the default column (A in this case) or should restore the sorting according to the last column clicked.

In UML statecharts, the first option (returning to super state's initial state) is the default. The second option (returning to the super state's state as it was before leaving) can be indicated by marking the super state as a History state, labeling it with a circled H. In both cases, if this behavior is important and requires testing, an extra path (scenario) is needed to verify the super state returns to the correct state after having been exited from a non-initial state.

And-states. Today's Web applications typically show a number of in-

Figure 17. State diagram for phone catalog.



dependent widgets, such as a contact list in one and an array of pictures in another. These widgets correspond to little independent state machines that are placed on one page.

In UML statecharts, such states can be described by *orthogonal regions* (also called AND-states), as shown in Figure 15. The Figure shows a Portfolio state, which consists of both a sortable table and an Upload button to add items. These can be used independently, as indicated by the two halves of the Portfolio state separated by the dashed line. The upload dialog is modal, which is why it is outside the Portfolio class. After uploading, the table remains sorted as it was, which is why it is labeled with the H.

Such orthogonal regions can be used to represent multiple little state machines present on one page. State transitions in these orthogonal regions may come from user interaction. They can also be triggered through server

events (over Web sockets) such as push notifications for new email and stock price adjustments, among others.

From a testing perspective, orthogonal regions are in principle independent and therefore can be tested independently.

Like OR-states, AND-states can be expanded, in this case to make all possible interleavings explicit. This blows up the diagram considerably and, hence, the number of potential test cases. While testing a few of such interleavings explicitly makes sense and is doable, testing all of them calls for automated test generation.

State-based stories. Last but not least, states and state diagrams can also be helpful when describing requirements with user stories and acceptance scenarios. For example, there is a natural fit with the story format proposed by technology consultant Dan North.⁹ Such stories consist of a general narrative of the form “as a... I want... so that...”, followed by a number of acceptance scenarios of the form “given... when... then”.

In many cases, these acceptance scenarios can be simply mapped to testing a single transition in the state diagram. The scenario then takes the form:

Given I have arrived in some state

When I trigger a particular event

Then the application conducts an action

And the application moves to some other state.

Thus, the state objects allow the API to interact with the state machine as suggested by these acceptance scenarios. A single test case moves from one state to another, and a full feature is described by a number of acceptance test cases navigating through the state machine, meanwhile checking how the application behaves.

AngularJS PhoneCat Example

As a complete example of testing with WebDriver and state objects, consider the AngularJS PhoneCat tutorial (<https://docs.angularjs.org/tutorial>). A screen shot of the PhoneCat application in action is shown in Figure 16. It comes with a test suite written in Protractor, which is the WebDriverJS¹¹ extension tailored toward AngularJS applications.

The application consists of a simple

list of phones that can be filtered by name and sorted alphabetically or by age. Clicking on one phone leads to full details for the given phone type.

The WebDriverJS test suite provided with the tutorial consists of three test cases for each of the two views (phone list and phone details), as well as one test for the opening URL, for a total of seven test cases.

The test suite in the original tutorial does not use page (or state) objects. To illustrate the use of state objects, I've rewritten the PhoneCat test suite to a state object-based test suite, which is available from my PhoneCat fork on GitHub (<https://github.com/avan-deursen/angular-phonecat/pull/1>).

The state diagram I used for the PhoneCat application is shown in Figure 17. It leads to two state objects (one for each view). These state objects can be used to express the original set of scenarios. Furthermore, the state diagram calls for additional cases, for example for the 'sort-newest' transition not covered in the original test case.

The figure also makes clear there is no direct way to get from Phone Details to the Phone List. Here the browser's back button is an explicit part of the interaction design, which is why the <<back>> annotation was added to the corresponding transition. (Note this is the only edge with this property: clicking "Back" after any other transition while in the Phone List state exits the application, as per AngularJS default behavior).

Since the back button is essential for navigating between the two views, the state-based test suite also describes this behavior through a scenario.

Lastly, as the Protractor and WebDriverJS APIs are entirely based on asynchronous JavaScript promises,¹¹ the state object implementations are asynchronous as well. For example, the Phone List state object offers a method that "schedules a command to sort the list of phones" instead of blocking until the phones are sorted. In this way, the actual scenarios can chain the promises together using, for example, the `then` promise operator.

AngularJS in production. Most of the figures presented in this article are based on diagrams created for a Web application developed for a Delft University of Technology spinoff company.

The application lets users register, log in, upload files, analyze and visualize them, and inspect analysis results.

The application's end-to-end test suite uses state objects. It consists of about 25 state objects and 75 scenarios. Like the PhoneCat test suite, it uses Protractor and consists of about 1,750 lines of JavaScript.

The end-to-end test suite is run from a TeamCity (<https://www.jetbrains.com/teamcity/>) continuous integration server, which invokes about 350 back-end unit tests, as well as all the end-to-end scenarios upon any change to the back end or front end.

The test suite has helped fix and find a variety of bugs related to client-side caching, back-button behavior, table sorting, and image loading. Several of these problems were a result of incorrect data bindings caused by, for example, typos in JavaScript variable names or incomplete rename refactorings. The tests also identified back-end API problems related to incorrect server configurations and permissions (resulting in, for example, a 405 and an occasional 500 HTTP status code), as well as incorrect client / server assumptions (the JavaScript Object Notation returned by the server did not conform to the front end's expectations).

Conclusion

When doing end-to-end testing of a Web application, use states to drive the tests:

- Model interactions of interest as small state machines.
- Let each state correspond to a state object.
- For each state include a self-check to verify the browser is indeed in that state.
- For each transition, write a scenario conducting self-checks on the original and target states, and verify the effects of the actions on the transition.
- Use the transition tree to reason about state reachability and transition coverage.
- Use advanced statechart concepts such as AND-states, OR-states, and annotations to keep your diagrams concise and comprehensible.
- Consider specific paths through the state machine that may be error prone; if you already have state objects for the states on that path, testing the behavior

along that path should be simple.

► Exercise the end-to-end test suite in a continuous integration server to spot integration problems between HTML, JavaScript, and back-end services.

As with page objects, the details of the browser interaction are encapsulated in the state objects and hidden from the test scenarios. Most importantly, the state diagrams and corresponding state objects directly guide you through the overall process of test-suite design. C

Related articles on queue.acm.org

Rules for Mobile Performance Optimization
Tammy Everts
<http://queue.acm.org/detail.cfm?id=2510122>

Scripting Web Service Prototypes
Christopher Vincent
<http://queue.acm.org/detail.cfm?id=640158>

Software Needs Seatbelts and Airbags
Emery D. Berger
<http://queue.acm.org/detail.cfm?id=2333133>

References

1. AngularJS. ngInclude directive; <https://docs.angularjs.org/api/ng/directive/ngInclude>.
2. Antoniol, G., Briand, L.C., Di Penta, M. and Labiche, Y. A case study using the round-trip strategy for state-based class testing. In *Proceedings 13th International Symposium on Software Reliability Engineering*. IEEE (2002), 269–279.
3. Binder, R.V. *Testing Object-oriented Systems*. Addison-Wesley, Reading, PA, 1999, Chapter 7.
4. Fowler, M. PageObject, 2013; <http://martinfowler.com/bliki/PageObject.html>.
5. Harel, D. 1987. Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 8(3): 231–274.
6. Horrocks, I. *Constructing the User Interface with Statecharts*. Addison-Wesley, Reading, PA, 1999.
7. Leotta, M., Clerissi, D., Ricca, F., Spadaro, C. Improving test suites maintainability with the page object pattern: An industrial case study. In *Proceedings of the Testing: Academic and Industrial Conference—Practice and Research Techniques*. IEEE (2013), 108–113.
8. Mesbah, A., van Deursen, A. and Roest, D. Invariant-based automatic testing of modern Web applications. *IEEE Transactions on Software Engineering* 38, 1 (2012) 35–53.
9. North, D. What's in a story?; <http://dannorth.net/whats-in-a-story/>.
10. Selenium. Page Objects, 2013; <https://github.com/SeleniumHQ/selenium/wiki/PageObjects>.
11. Selenium. Promises. In *WebDriverJS User's Guide*, 2014; <https://code.google.com/p/selenium/wiki/WebDriverJs#Promises>.
12. SeleniumHQ. WebDriver; <http://docs.seleniumhq.org/projects/webdriver/>.

Acknowledgments

Thanks to Michael de Jong, Alex Nederlof, and Ali Mesbah for many good discussions and for giving feedback on this post. The UML diagrams for this post were made with the free UML drawing tool UMLet (version 13.2).

Arie Van Deursen is a professor at Delft University of Technology where he leads the Software Engineering Research Group. To help bring his research into practice, he co-founded the Software Improvement Group in 2000 and Infotron in 2010.

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

Cloud computing for computer scientists.

BY DANIEL C. WANG

From the EDVAC to WEBVACs

BY NOW EVERYONE has heard of cloud computing and realized it is changing how traditional enterprise IT and emerging startups are building solutions for the future. Is this trend toward the cloud just a shift in the complicated economics of the hardware and software industry, or is it a fundamentally different way of thinking about computing? Having worked in the industry, I can confidently say it is both.

Most articles on cloud computing focus too much on the economic aspects of the shift and miss the fundamental changes in thinking. This article attempts to fill the gap and help a wider audience better appreciate some of the more fundamental issues related to cloud computing. Much of what is written here should not be earth shattering to those working with these systems day to day, but the article may encourage

even expert practitioners to look at their day-to-day issues in a more nuanced way.

Here are the key points to be covered:

- ▶ A cloud computer is composed of so many components that systematically dealing with the failure of those components is an essential factor to consider when thinking about the software that runs on a cloud computer or interacts with one.

- ▶ A common architectural pattern used to deal with transient failures is to divide the system into a pure computational layer and a separate layer that maintains critical system state. This provides reliability, scalability, and simplicity.

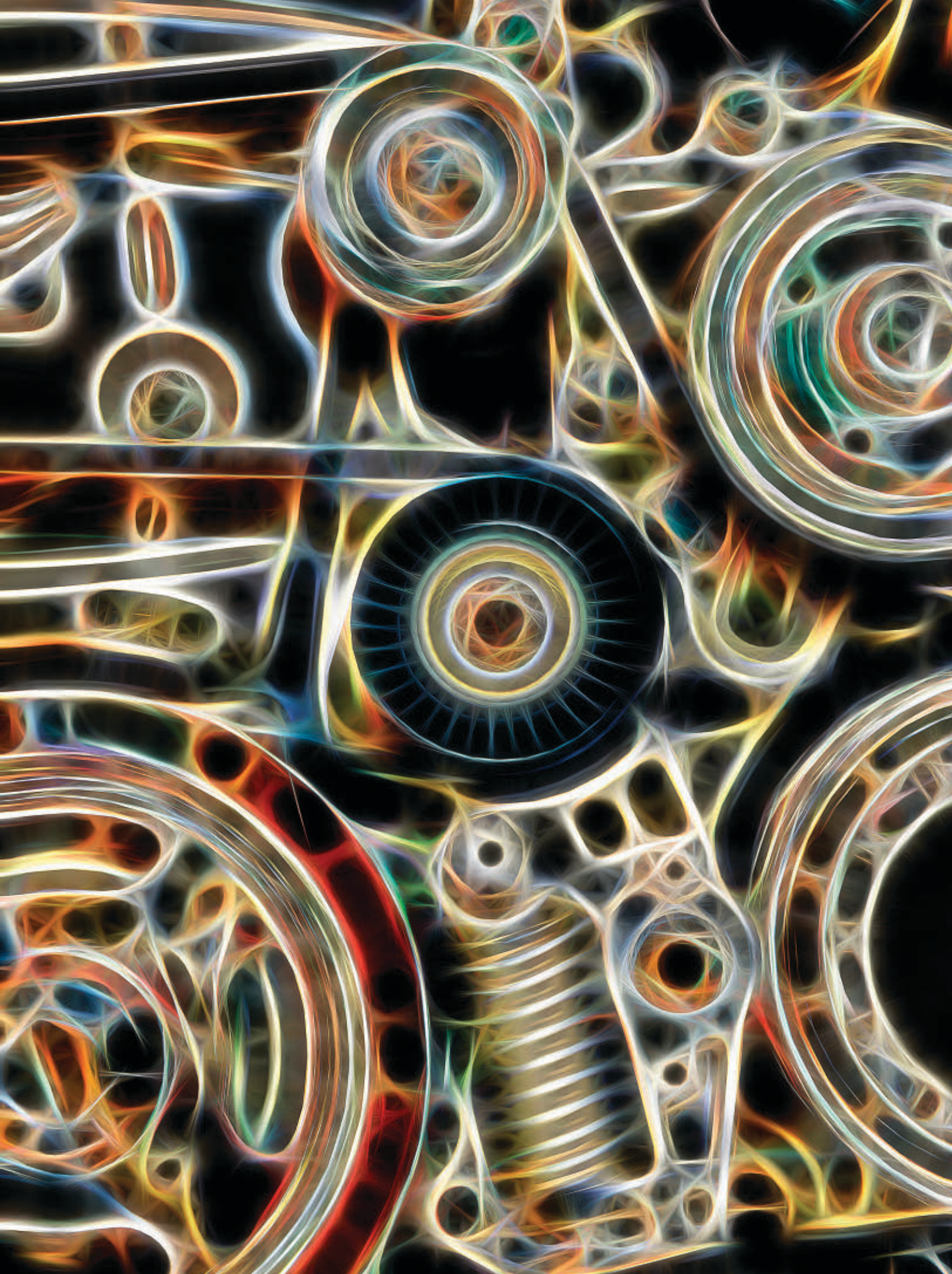
- ▶ A well-established existing best practice is to have systems expose idempotent interfaces so simple retry logic can be used to mask most transient failures.

- ▶ Simple analytic techniques can allow quantitative statements about various retry policies and compare how they impact reliability, worst-case latency, and average-case latency under an idealized failure model.

The first point about dealing with failure may seem new to many who are now hosting even small applications on large multitenant cloud computers in order to benefit from economies of scale. This is actually a very old issue, however, so the discussion should begin not by talking about the latest trends but by going back to the early years of the electronic computer.

Lessons from EDVAC and Von Neumann

In 1945 John von Neumann described the computational model of the first fully electronic stored program computer. This was a side effect of him acting as a consultant with ENIAC inventors John Mauchly and J. Presper Eckert. Although he was not the originator of many of the key ideas, his name is associated with the design approach, and the von Neumann architecture was soon a standard design



used for building electronic computers. The original EDVAC (electronic discrete variable automatic computer) draft report⁶ contains these interesting passages from von Neumann:

1.4 The remarks of 1.2 on the desired automatic functioning of the device must, of course, assume it functions faultlessly. Malfunctioning of any device has, however, always a finite probability—and for a complicated device and a long sequence of operations it may not be possible to keep this probability negligible. Any error may vitiate the entire output of the device. For the recognition and correction of such malfunctions intelligent human intervention will in general be necessary.

However, it may be possible to avoid even these phenomena to some extent. The device may recognize the most frequent malfunctions automatically, indicate their presence and location by externally visible signs, and then stop. Under certain conditions it might even carry out the necessary correction automatically and continue...

3.3 In the course of this discussion the viewpoints of 1.4, concerned with the detection, location, and under certain conditions even correction, of malfunctions must also receive some consideration. That is, attention must be given to facilities for checking errors. We will not be able to do anything like full justice to this important subject, but we will try to consider it at least cursorily whenever this seems essential.⁵

The practical problems that concerned von Neumann and the designers of the EDVAC in 1945 were the reliability of vacuum tubes and the main memory stored with mercury delay lines. (A modern hard drive is an amazing electromechanical device as well, which finally is starting to be replaced with solid-state memory.) The invention of the transistor, integrated circuit, and error-correcting codes makes von Neumann's concerns seem quaint today. Single-bit errors and even multi-bit errors in computer systems, while still possible, are sufficiently rare that these problems are considered inessential. The ability to consider failure of system components, however, can no longer be ignored with the advent of

cloud computers that fill acres of space with commodity servers.

Murphy's Law Triumphs over Moore's Law

A cloud computer is composed of so many components, each with a finite probability of failure, that the probability of all the components running without error at any point in time is close to zero. Failure and automated recovery are hence essential areas of concern not only at the hardware layer but also with the software components. In short, we are at the point where Murphy's Law has conquered Moore's Law. Assuming all the components make a system work flawlessly is a luxury that is no longer possible. Fortunately, techniques for handling bit-level corruptions can be adjusted and scaled to cloud computers, so most bit-level errors can be detected, if not fixed. The types of failures we are worried about, however, are those of a server or whole groups of servers. We are also at a point where the rates of certain failures are so high that von Neumann's suggestion that the system simply detect the error and wait for a human operator to intervene is no longer economically sensible.

You might ask if we can work harder to build more reliable systems, but when the reliability of your main power supply is inversely proportional to the density of wire-eating squirrels in your region or the probability that a worker will drop an uninsulated wrench into a power-distribution cabinet, it is difficult to imagine a cost-effective and systematic approach to address the reliability of data centers that commonly house as many as 100,000 servers.

From EDVAC to WEBVAC

One very popular approach to dealing with frequent server-level failures in a data center is to decompose the system into one or more tiers of servers that process requests on a best-effort basis and store any critical application state in a dedicated storage tier. Typically, there is a request load balancer in front of each tier so that the individual servers in the tier can fail and have requests rerouted automatically. The key aspect of this design is a complete separation between long-term system state and computation. This is the same separation that exists

between the processor and memory in the EDVAC design. For lack of a better term let's call these systems WEBVACs, for Worldwide Elastic Big Very Automated Clusters.

These WEBVACs are not conceptually different from the farms of Web servers seen today in a traditional data centers. WEBVACs use a proven architecture that provides resiliency, scalability, and a very familiar programming model based on stateless HTTP requests. The chief innovation is the degree and ease of configurability, as well as elasticity and scale. One important feature of EDVAC that distinguished it from earlier computers such as ENIAC was that EDVAC executed a program that was stored in its memory as data, while ENIAC was programmed by physically rewiring it for different problems. Figure 1 shows the conceptual similarity to both WEBVACs and the EDVAC.

Like EDVAC, modern cloud computers allow for the automatic configuration of a complete server farm with a few simple artifacts. This eliminates the need for tedious and error-prone manual configuration of servers, as is often done in more traditional systems.

Building a reliable storage tier that meets the needs of the compute tier is a challenging task. Requests in the storage tier need to be replicated across several servers using complex distributed consensus protocols. There is a wide range of approaches to building storage tiers, as well as a great diversity in their APIs and consistency models. (It is difficult to do justice to this topic in the limited space here, so see the additional reading suggestions at the end of this article.) In the end, however, the storage tier is just an abstraction that is callable from the compute tier. The compute tier can rely on the guarantees provided by the storage tier and therefore uses a much simpler programming model.

This simpler programming model, in which all the important state of the system is stored in a generic storage tier, also simplifies disaster-recovery scenarios since simple backup and restore of the storage tier is often sufficient to restore an entire system into a working state. Well-designed systems have asynchronous continuous backup of the storage tier to a replica in a physically different location. This

location needs to be close enough that data can be efficiently and cost-effectively replicated, but distant enough that the probability of it encountering the same “act of God” is low. (Putting both your primary and backup data centers near the same earthquake fault line is a bad idea.)

Since the backup is asynchronous, failover to the replica may incur some data loss. That data loss, however, can be bounded to acceptable and well-defined limits that come into play only when an act of God may cause the complete destruction of the primary system. Carefully determining the physical location of your data centers is the first case where there is a need to treat failure in an end-to-end way. This same end-to-end focus on failure is also important in the design and implementation of software running on and interacting with the system.

How to Design Interfaces for WEBVACs

WEBVACs ultimately provide APIs that allow desktop computers, mobile devices, or other WEBVACs to submit requests and receive responses to those requests. In any case, you end up with two agents that must communicate with each other via some interface over an unreliable channel. Reusing traditional designs from client-server systems or standard RPC

(remote procedure call) methods is not the best approach. Andrew Tanenbaum and Robbert van Renesse⁴ describe some common pitfalls when doing naïve refactoring of code not designed for distributed scenarios, which are generally applicable to the APIs here as well. One particular problem they call out is the 2AP (two-army problem), which demonstrates it is impossible to design a fully reliable method for two agents to reach consensus over an unreliable channel that may silently drop messages.

This is a restricted version of the more general problem of dealing with Byzantine failure, where the failure does not include data corruption. As a consequence, there is simply no way of building a system that can process any request with 100% reliability if the channel itself is unreliable. The 2AP

result, however, does not rule out protocols that asymptotically approach 100% reliability. A simple solution is continually transmitting a request up to some finite bound until some acknowledgment is received. If the error rate of the channel is fixed and failures are independent, then the likelihood of success increases exponentially with the number of transmissions.

In a large data center, not only is the communication between servers unreliable, but also the servers themselves are prone to failure. If a server in the compute tier fails, then a request that targeted it can be quickly rerouted to an equivalent compute server. The process of rerouting the request is often not fully transparent and the request is lost during rerouting, because the routing logic cannot immediately detect the server failure

Figure 1. EDVAC and WEBVAC.

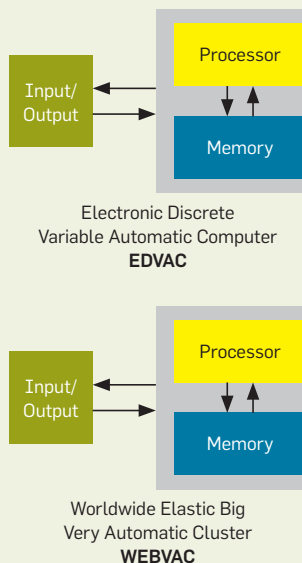


Figure 2. Simple interface used to enumerate files remotely.

```
enum Result {
    Ok,          /* Completed without errors. */
    NoMore,      /* No more names left to enumerate. */
    Fault        /* Message lost in transit or unknown failure. */
};

/* Moves cursor before first element in list of files. */
Result SetCursorToStart();

/* Get the current file name pointed to by the cursor.
 * Returns NoMore if the cursor is moved past the last name.
 */
Result GetCurrentFileName(char fileName[MAXLENGTH]);

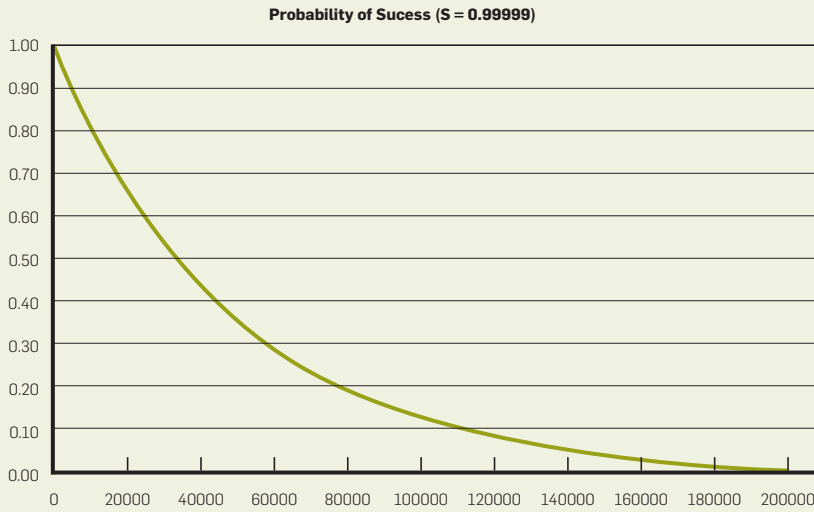
/* Move the cursor to the next file name.
 * Returns NoMore if there is none.
 */
Result MoveToNextFileName();
```

Figure 3. A naïve function to enumerate files.

```
Result ListFilesStopAtAnyFault() {
    char fileName[MAXLENGTH];
    Result res;
    res = SetCursorToStart();
    if (res != Ok) { return res; }
    printf("Start\n");
    for (;;) {
        res = MoveToNextFileName();
        if (res == NoMore) { break; }
        if (res != Ok) { return res; }
        res = GetCurrentFileName(fileName);
        printf("File: %s", fileName);
    }
    printf("End\n");
    return Ok;
}
```

Figure 4. Probability of success with five nines.

Request Success Rate (S)	Number of files (N)	Requests Sent (M)	Probability of Success
0.99999	N	$M = 2 \cdot (N + 1)$	S^M
0.99999	1	4	0.99996
0.99999	10	22	0.99978
0.99999	100	202	0.99798
0.99999	1,000	2,002	0.98017
0.99999	10,000	20,002	0.81871



or because the server was in the middle of processing a request when it failed. These lost requests appear to users as transient faults.

In the context of cloud computing, therefore, the observed request failure rate is really the combined error rate of the communication channel and the failure rate of the servers involved in the computation. Rather than reasoning about the individual failure rates of several components, you can make the simplifying assumption that a system of two unreliable agents communicating over an unreliable channel is equivalent to two idealized reliable agents communicating over an unreliable channel whose failure rate is increased appropriately to account for the failure of either of the original unreliable agents. An extended example illustrates this in more detail in the following section.

Enumerating a set of files over an unreliable channel. Figure 2 shows a simple interface definition in ANSI C that can be used to enumerate a set of file names. The interface has been exposed without careful consideration for failure, beyond the introduction of a new status code `Fault`, which indicates a failure likely caused by unreliable delivery. Assume that calling any one of these functions sends a request and waits synchronously for a response. The assumption is that the `Fault` status is returned if no response to a request is received after some fixed timeout.

Figure 3 illustrates a simple client-side function that attempts to enumerate all the files but returns immediately on the first `Fault` received by any call to the primitive functions in Figure 2.

You want to estimate the probability this function will return `Ok` under the assumption that calling any of the three functions mentioned earlier has a success rate of 0.99999 (S), which is to say that on average one out of a million invocations of the functions returns `Fault`. First you need to compute how many requests (M) are required to enumerate N files. Inspection of the code reveals that M is equal to

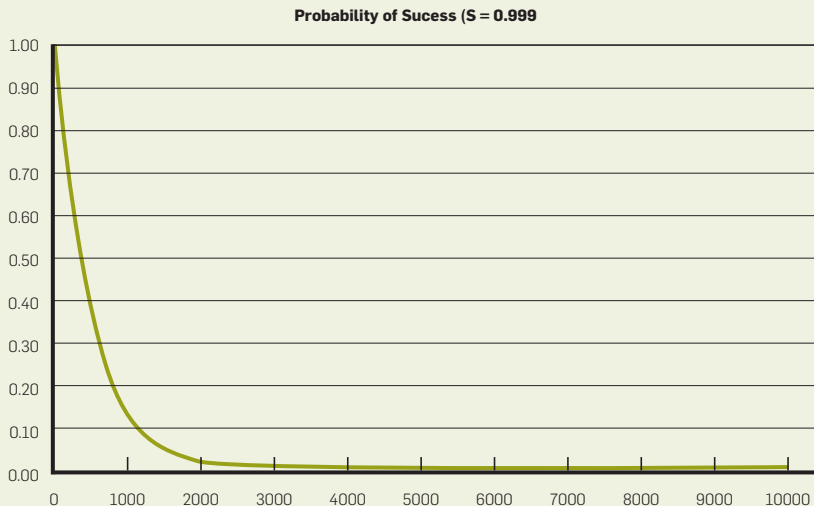
$$1 + 2 \cdot N + 1$$

which can be simplified to

$$2 \cdot (N + 1).$$

Figure 5. Probability of success with three nines.

Request Success Rate (S)	Number of files (N)	Requests Sent (M)	Probability of Success
0.999	N	$M = 2 \cdot (N + 1)$	S^M
0.999	1	4	0.99600
0.999	10	22	0.97822
0.999	100	202	0.81701
0.999	1,000	2,002	0.13492
0.999	10,000	20,002	$2 \cdot 10^{-9}$



Since the function fails immediately on any fault, the probability of no faults is simply the probability that all the requests sent succeed, which is S^M assuming failures are uniformly distributed. For purposes of this analysis, let's assume failure is independent and uniformly distributed. This simplifying assumption allows a comparison of the trade-offs of various approaches under equivalent ideal failure models. In practice, however, the distribution of failures is typically neither uniform nor completely independent. The results are summarized in Figure 4.

Depending on the workload characteristics, this success rate for the first attempt at listing files may be acceptable. In this example, the success rate of 0.99999 (a five-nines success rate results in fewer than 5.3 minutes of downtime a year for a continuously running system) is extremely high and typically can be achieved only with significant investment in expensive hardware infrastructure. A more realistic error rate would be 0.999 (a three-nines success rate results in fewer than 8.8 hours of downtime a year for a continuously running system), which is more typically seen with commodity components. A three-nines success rate produces the graph and table of values in Figure 5.

Clearly, a 3% failure rate for enumerating 10 files is not a usable system. You can improve the probability of success by simply retrying the whole function, but this is not only inefficient, but also, for large N , the success rate is so low that it would require an unreasonable number of retries. If the probability of the function `ListFilesStopAtAnyFault` enumerating N files successfully is

$$LS(N) = S^{2 \cdot (N+1)}$$

then the probability of failure is

$$LF(N) = 1 - LS(N).$$

The probability that after, at most, K retries the function succeeds is the same as the probability

$$1 - LF(N)^K$$

that is the complement of the probability that all invocations fail. For this discussion, if the probability of success is

Figure 6. Simple retry wrappers over remote primitives.

```
#define MAX_RETRIES 3
Result SetCursorToStartWithRetry() {
    Result res;
    for (int i = 0; i < MAX_RETRIES; i++) {
        res = SetCursorToStart();
        if (res != Fault) { return res; }
    }
    return Fault;
}

Result GetCurrentFileNameWithRetry(
    char fileName[MAXLENGTH]) {
    ...
}

Result MoveToNextFileNameWithRetry() {
    ...
}
```

Figure 7. Enumerating files using retry wrappers.

```
Result ListFilesWithRetry() {
    char fileName[MAXLENGTH];
    Result res;
    res = SetCursorToStartWithRetry();
    if (res != Ok) { return res; }
    printf("Start\n");
    for (;;) {
        res = MoveToNextFileNameWithRetry();
        if (res == NoMore) { break; }
        if (res != Ok) { return res; }
        res = GetCurrentFileNameWithRetry(fileName);
        printf("File: %s", fileName);
    }
    printf("End\n");
    return Ok;
}
```

Figure 8. An idempotent interface to enumerate files remotely.

```
typedef int tok_t;
/* Get start token with cursor before
first element in list of files. */
Result GetStartToken(
    tok_t *init);

/*
 * Get the current file name pointed to
 * by the cursor relative to a state token.
 * Returns NoMore if the cursor is moved past
 * the last name.
 */
Result GetCurrentFileNameWithToken(
    tok_t curent,
    char fileName[MAXLENGTH]);

/* Move the given a token return the next state
 * token with the cursor advanced. Returns NoMore
 * if there is none.
 */
Result MoveToNextFileNameWithToken(
    tok_t curent,
    tok_t *next);
```

at least 0.999, when N is 100, you must retry, on average, five times; when N is 1,000, the number of retries is at least 50 to get a three-nines success rate; for $N = 10,000$, the number is close to 3 billion. The smarter approach is to keep `ListFilesStopAtAnyFault` from

immediately failing on any single fault.

This can be accomplished by creating simple wrapper functions that add some basic retry logic over the original primitives so there is a new set of more robust primitives, as shown in Figure 6.

Production code would likely in-

clude an exponential back-off that delays each retry with an exponentially increasing time delay. This avoids the so-called “thundering herd” problem when many clients are simultaneously trying to recover from a network partition to a given server. For simplicity, this discussion will ignore it. Assuming a success rate of 0.999 for the underlying primitives, performing three simple retries makes the probability of each of these returning without a fault

$$1 - (1 - 0.999)^3$$

or 0.999999999 (nine nines). Figure 7 shows how you can now write a new routine that uses these more reliable wrappers.

Now you can evaluate the reliability of the function `ListFilesWithRetry`, but instead of computing this with respect to primitive requests, you compute it with respect to the number of times each request wrapper is called (see Table 1).

Now that each wrapper has a nine-nines success rate, the overall success rate for this function, even when $N = 10,000$, is more than 0.9999 (a four-nines success rate results in fewer than 53 minutes of downtime a year for a continuously running system). There is still a nonzero chance this function will return `Fault`, so this has not solved the 2AP but has significantly increased the likelihood that the system will make progress. The insertion of retries, of course, increases the overall latency when there are errors, and, with a reasonable model of latency, the expected time for enumerating N files assuming a specific request failure rate can be computed. The latency impact of these changes is discussed later.

Astute readers should notice a fatal flaw in the code. The function will continue to enumerate under the presence of request failures, but the naïve addition of retries will cause files to be skipped when there are failures. Specifically, this wrapper function may cause files to be skipped:

```
Result MoveToNextFileNameWithRetry();
```

The fundamental issue here is that the underlying primitive request `MoveToNextFileName` is not idempotent—one invocation of it is not

Table 1. Probability of success using retry wrappers.

Wrapper Success Rate (W)	Number of files (N)	Wrappers Called (C)	Probability of Success
$1 - (1 - 0.999)^3$	N	$C = 2 \cdot (N + 1)$	W^C
0.999999999	1	4	0.999999996
0.999999999	10	22	0.999999978
0.999999999	100	202	0.999999798
0.999999999	1,000	2,002	0.999997998
0.999999999	10,000	20,002	0.999979998

Figure 9. Enumerating files with idempotent retries.

```
Result ListFilesWithTokenAndRetry() {
    char fileName[MAXLENGTH];
    Result res;
    tok_t current;
    res = GetStartTokenWithRetry(&current);
    if (res != Ok) { return res; }
    printf("Start\n");
    for (;;) {
        res = MoveToNextFileNameWithTokenAndRetry(current, &current);
        if (res == NoMore) { break; }
        if (res != Ok) { return res; }
        res = GetCurrentFileNameWithTokenAndRetry(current, fileName);
        printf("File: %s", fileName);
    }
    printf("End\n");
    return Ok;
}
```

Figure 10. A less chatty idempotent interface.

```
/* Well known start token. */
tok_t StartToken = -1;

/* Given a token return the next state
 * token with the cursor advanced and
 * current file name. Returns NoMore if
 * there are no files.
 */
Result GetCurrentFileNameAndNextToken(
    tok_t current,
    char fileName[MAXLENGTH],
    tok_t *next);
```

Table 2. Expected latency of final protocol.

N	$L'_{\max}(N)$	$L'_{\text{slow}}(N)$	$L'_{\text{avg}}(N)$
1	6 seconds	2 seconds	22 milliseconds
10	33 seconds	11 seconds	121 milliseconds
100	5.05 minutes	1.68 minutes	1.11 seconds
1,000	50.5 minutes	16.7 minutes	11.0 seconds
10,000	8.33 hours	2.78 hours	1.83 minutes

observationally equivalent to multiple invocations of the function. Because of the 2AP, there is no way to have the server or client agree if the cursor has moved forward or not on a fault. The only way to resolve this issue is to make `MoveToNextFileName` idempotent.

There are a variety of techniques to do this. One way is to include sequence numbers to detect retries and have the server track these numbers. These sequence numbers now become important state that must be placed in the storage tier for every client in the system, and this can result in scalability issues. A more scalable approach is to use an opaque state token similar to how cookies are used in HTTP to offload state from the server to the client. The client can maintain the needed state rather than have the server track. This leaves the API shown in Figure 8, which includes only idempotent functions.

In this example the state token is simply an integer in a realistic system; it would more likely be a variable-length byte array that the system verifies is valid so that malicious clients cannot harm the system.

The retry wrappers `GetStartTokenWithRetry`, `GetStartTokenWithRetry`, and `MoveToNextFileNameWithTokenAndRetry` can also be defined as earlier.

We can adjust our function to use wrapped primitives over the idempotent API (see Figure 9).

The analysis performed earlier on the slightly buggy version that lacked the idempotent primitives is still valid, but now the function works correctly and reliably. When $N = 10,000$, the success rate is 0.999979998 (four nines).

Analyzing and Improving Latency

Because the only practical way to detect message loss between senders and receivers is via timeouts, it is important to document a time bound on how long either party should wait for a request to be processed. If a request is processed after that time bound has been exceeded, then consider the request failed. Services typically define an upper bound for each API function they support (for example, 99.9% of all requests will be successfully processed within one second). If no upper time bound is specified, a guaranteed 99.9%

success rate is somewhat meaningless; you may have to wait an infinite amount of time for any single request to complete successfully.

Determining a reasonable upper bound for a system can be quite complex. It can be estimated observationally by looking at the distribution of latencies across a large sample of requests and choosing the maximum, or by simply having the server include a watchdog timer with a clear upper bound that fails any request that exceeds the threshold. In either case, the worst-case bound is needed so that clients of the service can set timeouts appropriately, but these worst-case bounds typically are very conservative estimates of the actual average-case performance. An extended version of this article published in *ACM Queue* analyzes worst-case, average-case, and slow-case latency for the file-listing function.

Let's assume that every primitive request to the server has a worst-case successful latency of R_{max} and average time of R_{avg} , with these parameters we can analytically estimate how our retry policy impacts latency. There worst-case is based on a pathological worst-case scenario where the maximum number of failures and retries occur. This analysis is overly pessimistic, and we can instead use our average-case analysis to derive a slow-case estimate where the distribution of retries is the same as our average-case analysis, but we assume every request is as slow as possible.

Some readers may feel this final implementation is too "chatty" and a more efficient protocol could reduce server round-trips. Indeed, three functions in the API can be replaced with one (see Figure 10).

This protocol has a fixed globally known start token and a single function that returns both the current file name and next token in one request. There should be an expected improvement in latency, which can be seen by performing a latency analysis of the modified protocol depicted in Table 2.

The details of this latency analysis are in the full article, and use very elementary techniques of probability theory to analytically determine reasonable time out values for callers of the function to list files.

Conclusion

This article is a far from exhaustive survey about the many interesting issues surrounding cloud computing. The goal is to demonstrate the breadth of deep problems still to be solved. Some of the trailblazers who developed the electronic computer would be dumbfounded by the computation we now carry in our pockets. They would be equally surprised at how robustly some of their earliest ideas have stood the test of time. Taken in historical context, the modern WEBVAC should not be seen as the culmination of 70 years of human progress, but just the start of a promising future that we cannot imagine.

Acknowledgments

Special thanks to Gang Tan for encouraging me to write this article, and Steve Zdancewic for providing feedback. 

Related articles on queue.acm.org

Describing the Elephant: The Different Faces of IT as Service

Ian Foster and Steven Tuecke
<http://queue.acm.org/detail.cfm?id=1080874>

Lessons from the Floor

Daniel Rogers
<http://queue.acm.org/detail.cfm?id=1113334>

From EDVAC to WEBVACs

Daniel C. Wang
<http://queue.acm.org/detail.cfm?id=2756508>

References

1. Calder, B., Wang, J., Ogun, A. et al. Windows Azure Storage: A highly available cloud storage service with strong consistency. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles* (2011), 143–157; DOI=10.1145/2043556.2043571.
2. DeCandia, G., Hastorun, D. et al. Dynamo: Amazon's highly available key-value store. In *Proceedings of 21st ACM Symposium on Operating Systems Principles* (2007), 205–220; DOI=10.1145/1294261.1294281.
3. Ghemawat, S., Gobioff, H., Leung, S.-T. The Google file system. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, (2003), 29–43. DOI=10.1145/945445.945450; <http://doi.acm.org/10.1145/945445.945450>.
4. Tanenbaum, A.S., van Renesse, R. A critique of the remote procedure call paradigm. In *European Teleinformatics Conference Proceedings, Participants Edition* (1988), 775–783.
5. von Neumann, J. *First Draft of a Report on the EDVAC*. Technical Report, 1945. <https://web.archive.org/web/20130314123032/http://qss.stanford.edu/~godfrey/vonNeumann/vnedvac.pdf>.
6. Wikipedia. First draft of a report on the EDVAC; http://en.wikipedia.org/wiki/First_Draft_of_a_Report_on_the_EDVAC.

Daniel C. Wang has been working in the computing industry for more than 15 years. He works on the Azure Web Workload team. Any opinions in the article are his and not of his employer.

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

DOI:10.1145/2699415

The Quipper language offers a unified general-purpose programming framework for quantum computation.

**BY BENOÎT VALIRON, NEIL J. ROSS, PETER SELINGER,
D. SCOTT ALEXANDER, AND JONATHAN M. SMITH**

Programming the Quantum Future

THE EARLIEST COMPUTERS, like the ENIAC, were rare and heroically difficult to program. That difficulty stemmed from the requirement that algorithms be expressed in a “vocabulary” suited to the particular hardware available, ranging from function tables for the ENIAC to more conventional arithmetic and movement operations on later machines. Introduction of symbolic programming languages, exemplified by FORTRAN, solved a major difficulty for the next generation of computing devices by enabling specification of an algorithm in a form more suitable for human understanding, then translating this specification to a form executable by the machine. The “programming language” used for such specification bridged a semantic gap between the human and the computing device. It provided two important features: high-level abstractions, taking care of automated bookkeeping, and modularity, making it easier to reason about sub-parts of programs.

Quantum computation is a computing paradigm where data is encoded in the state of objects governed by the laws of quantum physics. Using quantum techniques, it is possible to design algorithms that outperform their best-known conventional, or classical, counterparts.

While quantum computers were envisioned in the 20th century, it is likely they will become real in the 21st century, moving from laboratories to commercial availability. This provides an opportunity to apply the many lessons learned from programming classical computing devices to emerging quantum computing capabilities.

Quantum Coprocessor Model

How would programmers interact with a device capable of performing quantum operations? Our purpose here is not to provide engineering blueprints for building an actual quantum computer; see Meter and Horsman¹³ for a discussion of that agenda. What we describe is a hypothetical quantum architecture in enough detail to cover how one would go about programming it.

Viewed from the outside, quantum computers perform a set of specialized operations, somewhat analogous to a floating-point unit or a graphics coprocessor. We therefore envision the quantum computer as a kind of coprocessor that is controlled by a classical computer, as shown schematically in Figure 1. The classical computer per-

» key insights

- Quantum computer science is a new discipline dealing with the practical integration of all aspects of quantum computing, from an abstract algorithm in a research paper all the way to physical operations.
- The programs written in a quantum programming language should be as close as possible to informal high-level descriptions, with output suitable for the quantum coprocessor model.
- Other important aspects of the quantum programming environment include automated offline resource estimates prior to deployment and facilities for testing, specification, and verification.

ILLUSTRATION BY FUTUREDELUXE



forms operations (such as compilation, conventional bookkeeping, correctness checking, and preparation of code and data) for the quantum unit. The quantum coprocessor performs

only the quantum operations (such as initializations, unitary operations, and measurements). This model of quantum computation is known as Knill's QRAM model¹¹ and is believed to ulti-

mately be the most likely realization of quantum computers.¹³

Certain hardware-intensive low-level control operations (such as quantum error correction) may optionally be integrated directly into the quantum unit. We envision the quantum unit containing a high-speed, specialized firmware in charge of such a low-level “quantum runtime.” The quantum firmware is specific to each physical realization of a quantum coprocessor, programmed separately off site. Although tightly dependent on the physical specifications of the particular hardware, the quantum firmware is independent of the algorithms to be run.

The source code of any quantum programs resides on the classical unit. Through a conventional classical compilation, it produces executable code to be run on the conventional computer. We envision the quantum coprocessor will communicate with its classical controller through a message queue on which the classical computer is able to send elementary instructions (such as “allocate a new quantum bit,” “rotate quantum bit x ,” and “measure quantum bit y ”). After an operation is performed, the classical computer can read the results from the message queue. In this model, the control flow of an algorithm is classical; tests and loops are performed on the classical device. Both classical and quantum data are first-class objects.

Via the message queue, the classical runtime receives feedback (such as the results of measurements) from the quantum unit. Depending on the algorithm, this feedback may occur only at the end of the quantum computation (batch-mode operation) or interleaved with the generation of elementary instructions (online operation). The possibility of online operation raises additional engineering challenges, as it requires the classical controller to be fast enough to interact with the quantum runtime in real time. On the other hand, many common quantum algorithms require only batch-mode operation. We assume a quantum programming model flexible enough to address either type of operation.

As with a conventional programming environment, we separate the logical data structures from their physical

Figure 1. Mixed computation in the quantum coprocessor model.

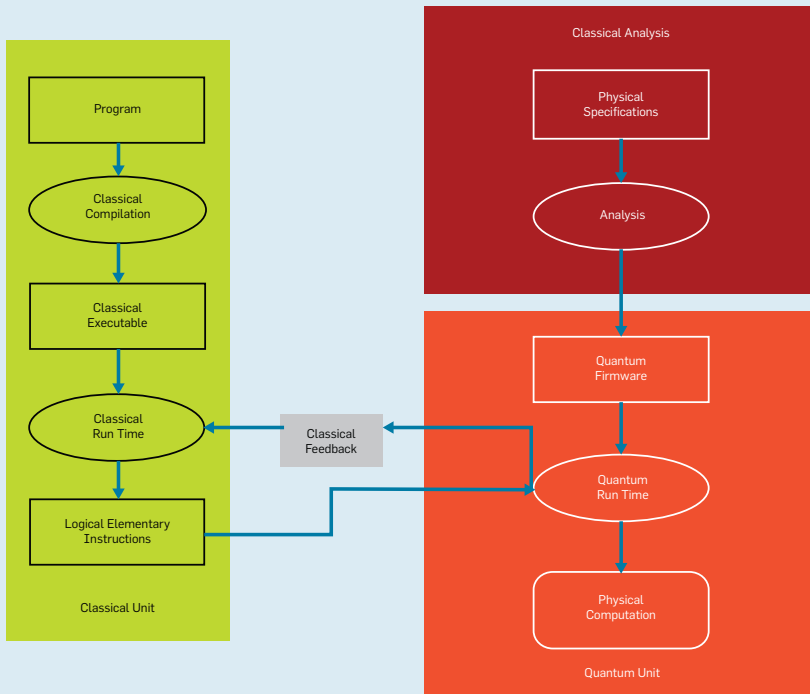


Figure 2. A quantum circuit.

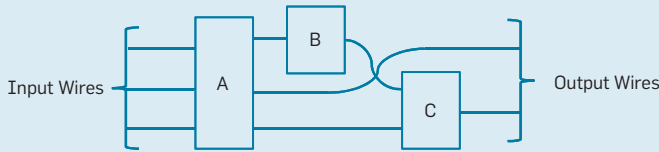
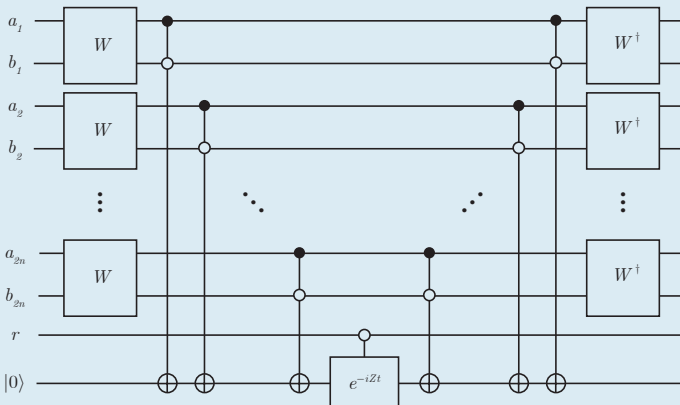


Figure 3. A quantum circuit fragment.



representation on the hardware. In our proposed paradigm, the algorithms are implemented at the logical level, but the quantum bits are physically encoded at the hardware level. The tasks of mapping logical quantum bits and operations to stable physical representations, and of applying suitable error correction, are left to the compiler and to the quantum firmware.

Describing Quantum Algorithms

To motivate the need for an expressive quantum programming language (QPL), we briefly consider some of the ways quantum algorithms are typically specified in the literature. A quantum algorithm generally consists of a mix of classical and quantum operations. The quantum parts of an algorithm are usually aggregated into quantum circuits, in which quantum gates are represented by boxes and quantum bits by wires, as in Figure 2. Some restrictions apply to circuits. For example, they cannot contain loops, so wires

must flow in only one direction. The gates can have multiple inputs and outputs. With the exception of gates corresponding to the creation and destruction (measurement) of quantum bits, elementary operations are always unitary transformations, implying they must have the same number of inputs and outputs.

A typical description of a quantum algorithm consists of one or more of the following pieces, which may be

specified at various levels of formality:

Mathematical equations. These can be used to describe state preparations, unitary transformations, and measurements. For example, Harrow et al.⁹ described a quantum algorithm for solving linear systems of equations. A certain subcircuit of the algorithm is defined as follows:

$$\sum_{t=1}^T \left(\begin{array}{l} |t+1\rangle\langle t| \otimes U_t \\ + |t+T+1\rangle\langle t+T| \otimes I \\ + |t+2T+1 \bmod 3T\rangle\langle t+2T| \otimes U_{3T+1-t}^\dagger \end{array} \right)$$

Figure 4. Inversion and repetition.

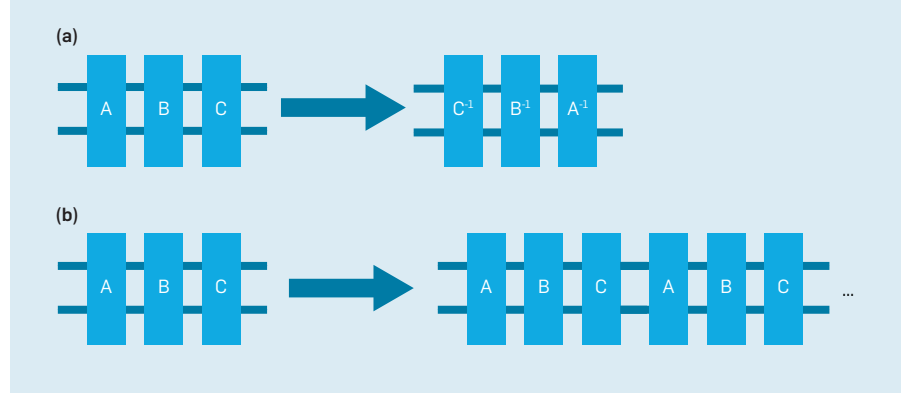


Figure 5. An initialize-run-measure loop.

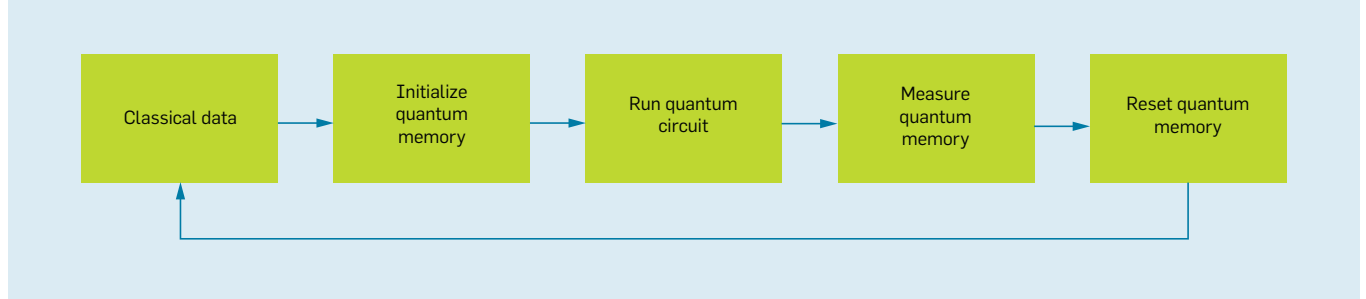


Figure 6. A circuit with feedback from intermediate measurements.

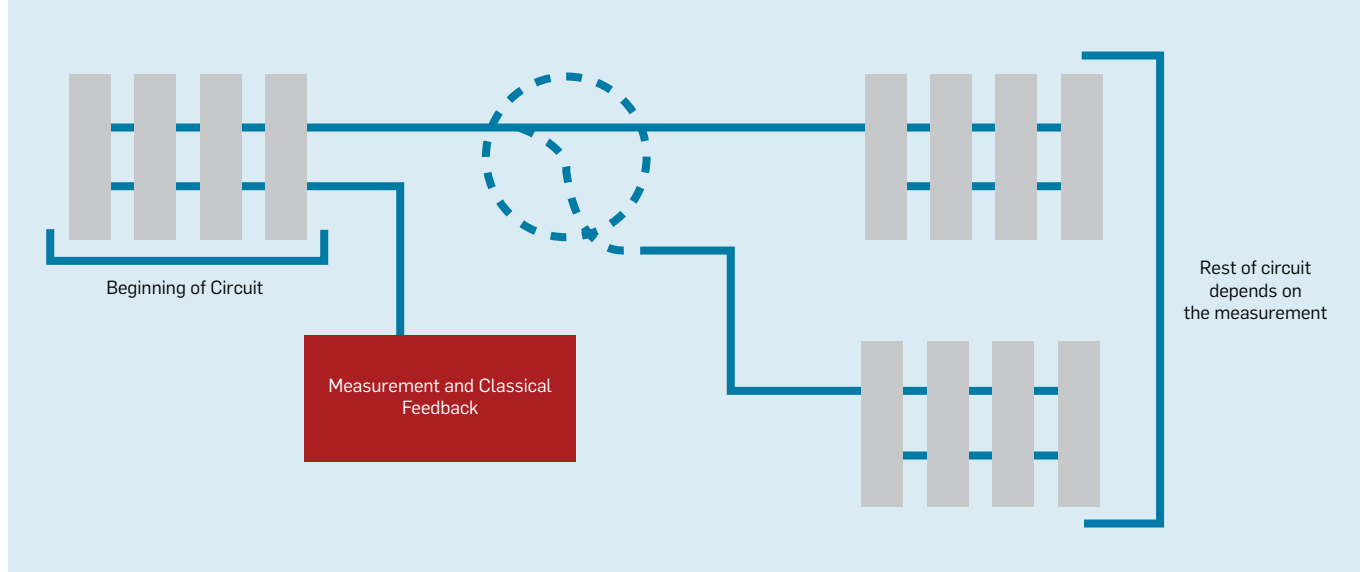
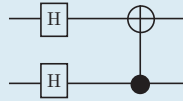


Figure 7. A procedural example.

```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```



Invocation of known quantum subroutines. Examples include the quantum Fourier transform, phase estimation, amplitude amplification, and random walks. For example, the algorithm in Harrow et al.⁹ asks to “decompose $|b\rangle$ in the eigenvector basis, using phase estimation”;

Oracles. These are classical computable functions that must be made reversible and encoded as quantum operations. They are often described at a very high level; for example, Burham et al.³ defined an oracle as “the truth value of the statement $f(x) \leq f(y)$ ”;

Circuit fragments. For example, the circuit in Figure 3 is from Childs et al.⁴ Note, strictly speaking, the figure describes a *family* of quantum circuits, parameterized by a rotation angle t and a size parameter n , as indicated by ellipses “...” in the informal circuit. In a formal implementation, this parameter dependency must be made explicit;

High-level operations on circuits. Examples include “inversion,” where a circuit is reversed, and “iteration,” where a circuit is repeated, as in Figure 4; and

Classical control. Many algorithms involve interaction between the classical unit and the quantum unit. This interaction can take the form of simple iteration, running the same quantum

circuit multiple times from scratch, as in Figure 5, or of feedback, where the quantum circuit is generated on the fly, possibly based on the outcome of previous measurements, as in Figure 6.

Requirements for QPLs

Ideally, a quantum programming language should permit programmers to implement quantum algorithms at a level of abstraction that is close to how one naturally thinks about the algorithm. If the algorithm is most naturally described by a mathematical formula, then the programming language should support such a description where possible. Similarly, if the algorithm is most naturally described by a sequence of low-level gates, the programming language should support this description as well.

The standard methods used to present many algorithms in the literature can therefore be taken as guidelines in the design of a language. Knill¹¹ laid out requirements for quantum programming, including:

Allocation and measurement. Make it possible to allocate and measure quantum registers and apply unitary operations;

Reasoning about subroutines. Permit reasoning about quantum subroutines, reversing a subroutine, and con-

ditioning a subroutine over the state of a quantum register; and

Building quantum oracles. Enable the programmer to build a quantum oracle from the description of a classical function.

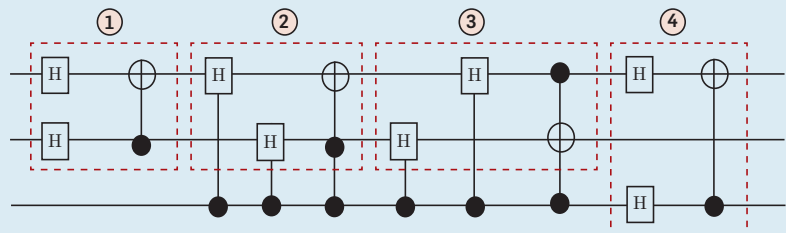
Our experience implementing^a quantum algorithms suggests some additional features that would be requirements for a quantum programming language:

Quantum data types. In classical languages, data types are used to permit the programmer to think abstractly about data instead of managing individual bits or words. For example, in most situations, a floating-point number is best viewed as a primitive data type supporting certain arithmetic operations, rather than an array of 64 bits comprising an exponent, a mantissa, and a sign. Likewise, many quantum algorithms specify richer data types, so the language should also provide these abstractions. For example, the Quantum Linear System Algorithm⁹ requires manipulation of quantum integers and quantum real and complex numbers that can be represented through a floating-point or fixed-precision encoding. Another example is Hallgren’s algorithm⁸ for computing the class number of a real quadratic number field. One type of data that occurs in this algorithm, and that must be put into quantum superposition, is the type of ideals in an algebraic number field;

a By “implementing” an algorithm, we mean realizing it as a computer program; we do not mean we have actually run the programs on a quantum computer, although we have run parts of the algorithms on quantum simulators.

Figure 8. A block structure example.

```
mycirc2 :: Qubit -> Qubit -> Qubit -> Circ (Qubit, Qubit, Qubit)
mycirc2 a b c = do
  ① mycirc a b
  with_controls c $ do
    ② mycirc a b
    ③ mycirc b a
    ④ mycirc a c
  return (a,b,c)
```



Specification and verification. In classical programming, there is a variety of techniques for ascertaining the correctness of programs, including compile-time type checking, runtime type checking, formal verification, and debugging. Among them, formal verification is arguably the most reliable but also the most costly. The availability of strong compile-time guarantees requires very carefully designed programming languages. Debugging is cheap and useful and therefore ubiquitous in classical-program development.

In quantum computation, the cost of debugging is likely to be quite high. To begin with, observing a quantum system can change its state. A debugger for a quantum program would therefore necessarily give incomplete information about its state when run on actual quantum hardware. The alternative is to use a quantum simulator for debugging. But this is not practical due to the exponential cost of simulating quantum systems. Moreover, it can be expected that the initial quantum computers will be rare and expensive to run and therefore that the cost of runtime errors in quantum code will initially be much higher than in classical computing. This shifts the cost-benefit analysis for quantum programming toward strong compile-time correctness guarantees, as well as formal specification and verification.

A quantum programming language should have a sound, well-defined semantics permitting mathematical specifications of program behavior and program correctness proofs. It is also beneficial for the language to

Figure 9. A circuit operator example.

```
timestep :: Qubit -> Qubit -> Qubit -> Circ (Qubit, Qubit, Qubit)
timestep a b c = do
  mycirc a b
  qnot c `controlled` (a,b)
  reverse_simple mycirc (a,b)
  return (a,b,c)
```

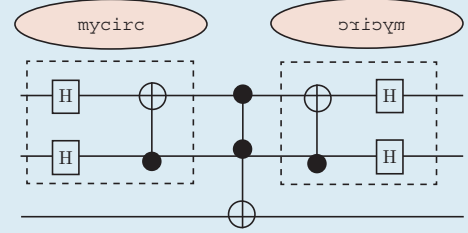


Figure 10. A circuit transformer example.

```
timestep2 :: Qubit -> Qubit -> Qubit -> Circ (Qubit, Qubit, Qubit)
timestep2 = decompose_generic Binary timestep
```

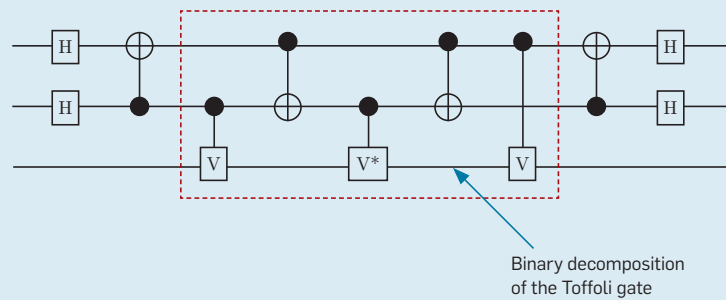


Figure 11. A functional-to-reversible translation example.

```
build_circuit
f :: [Bool] -> Bool
f as = case as of
  [] -> False
  [h] -> h
  h:t -> h 'bool_xor' f t

unpack template_f :: [Qubit] -> Circ Qubit
```

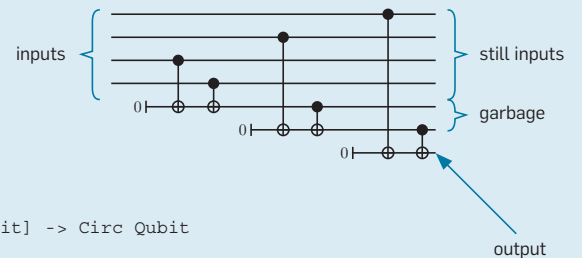


Figure 12. The circuit from Figure 11 made reversible.

```
classical_to_reversible :: (Datab a, QCData b) => (a -> Circ b) -> (a,b) -> Circ (a,b)
classical_to_reversible (unpack template_f)
```

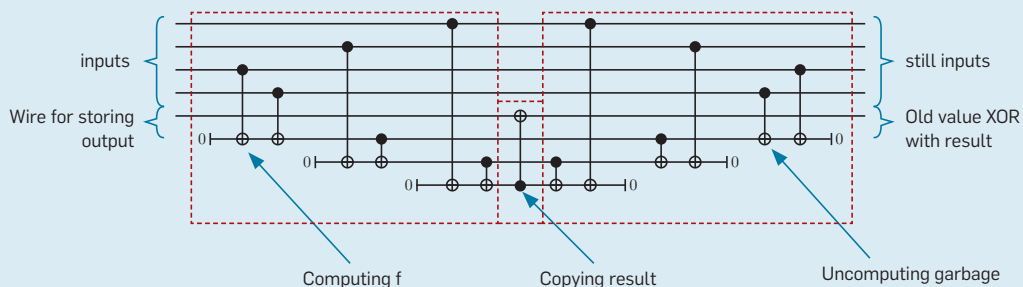


Figure 13. A procedural example.

```

import Quipper

w :: (Qubit,Qubit) -> Circ (Qubit,Qubit)
w = named_gate "W"

toffoli :: Qubit -> (Qubit,Qubit) -> Circ Qubit
toffoli d (x,y) =
  qnot d 'controlled' x .==. 1 .&&. y .==. 0

eiz_at :: Qubit -> Qubit -> Circ ()
eiz_at d r =
  named_gate_at "eiz" d 'controlled' r .==. 0

circ :: [(Qubit,Qubit)] -> Qubit -> Circ ()
circ ws r = do
  label (unzip ws,r) (("a","b"),"r")
  with_ancilla $ \d -> do
    mapM_ w ws
    mapM_ (toffoli d) ws
    eiz_at d r
    mapM_ (toffoli d) (reverse ws)
    mapM_ (reverse_generic w) (reverse ws)
    return ()

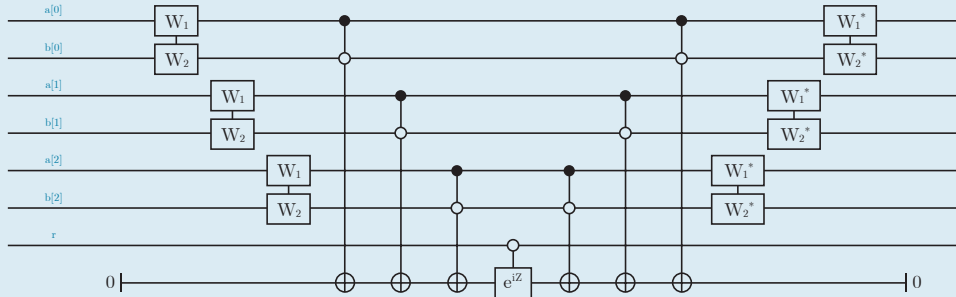
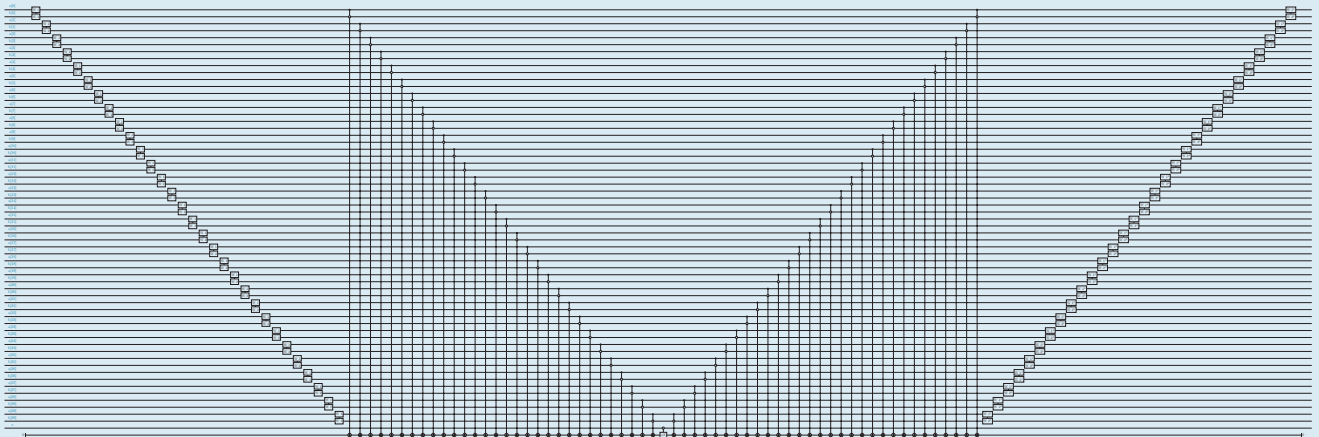
main = print_generic EPS circ (replicate 3 (qubit,qubit)) qubit

```

have a strong static type system that can guarantee the absence of most runtime errors (such as violations of the no-cloning property of quantum information);^b and

Resource sensitivity and resource estimation. At first, quantum computers will probably not have many qubits. The language should thus include tools to estimate the resources required for running a particular piece of code (such as number of qubits, number of elementary gates, or other rel-

^b The absence of cloning is already guaranteed by the physics, regardless of what the programming language does. However, one could similarly say the absence of an illegal memory access is guaranteed by a classical processor's page-fault mechanism. It is nevertheless desirable to have a programming language that can guarantee *prior to running the program* that the compiled program will never attempt to access an illegal memory location or, in the case of a quantum programming language, will not attempt to apply a controlled-not gate to qubits n and m , where $n = m$.

Figure 14. The circuit generated by the code in Figure 13, with three qubit pairs.**Figure 15. The circuit generated by the code in Figure 13, with 30 qubit pairs.**

evant resources) prior to deployment. One particular issue for language designers is how to handle quantum error correction. As the field advances, a decision must be made as to whether error correction should be exposed to the programmer (potentially allowing for optimization by hand) or whether it is more efficient to let the compiler or some other tool apply error correction automatically. Due to the potential for undesirable interactions between quantum error correction (which adds redundancy) and the optimization step of a compiler (which removes redundancy), the design and implementation of any quantum programming language must be aware of the requirements of quantum error correction.

Prior Work on QPLs

Several quantum programming languages have been developed by researchers around the world.⁵ Some, including van Tonder's quantum lambda calculus,¹⁸ are primarily intended as theoretical tools. The first quantum programming language intended for practical use was arguably Ömer's QCL,¹⁴ a C-style imperative language supporting "structured quantum programming." QCL provides simple registers but no high-level quantum data types. It could also benefit from greater support for specification and verifica-

Figure 16. The `calcRweights` function.

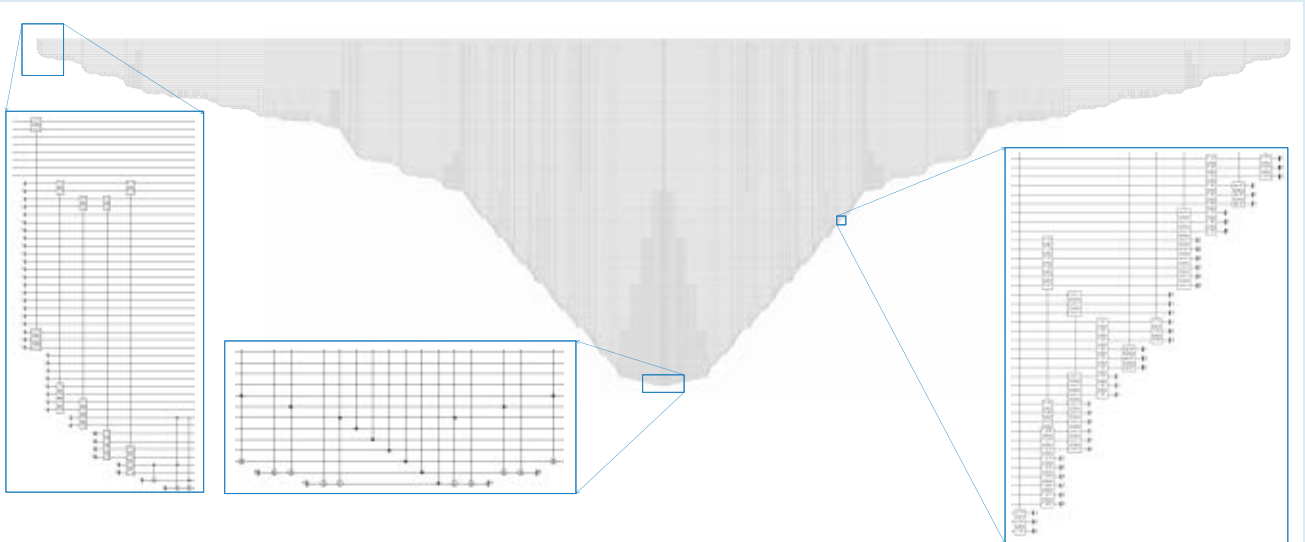
```
calcRweights y nx ny lx ly k theta phi =
  let (xc',yc') = edgetoxy y nx ny in
  let xc = (xc'-1.0)*lx - ((fromIntegral nx)-1.0)*lx/2.0 in
  let yc = (yc'-1.0)*ly - ((fromIntegral ny)-1.0)*ly/2.0 in
  let (xg,yg) = itoxy y nx ny in
  if (xg == nx) then
    let i = (mkPolar ly (k*xc*(cos phi)))*(mkPolar 1.0 (k*yc*(sin phi)))*
      ((sinc (k*ly*(sin phi)/2.0)) :+ 0.0) in
    let r = ( cos(phi) :+ (- k*lx) )*((cos (theta - phi))/lx :+ 0.0) in i * r
  else if (xg==2*nx-1) then
    let i = (mkPolar ly (k*xc*cos(phi)))*(mkPolar 1.0 (k*yc*sin(phi)))*
      ((sinc (k*ly*(sin phi)/2.0)) :+ 0.0) in
    let r = ( cos(phi) :+ (- k*lx) )*((cos (theta - phi))/lx :+ 0.0) in i * r
  else if ( (yg==1) && (xg<nx) ) then
    let i = (mkPolar lx (k*yc*sin(phi)))*(mkPolar 1.0 (k*xc*cos(phi)))*
      ((sinc (k*lx*(cos phi)/2.0)) :+ 0.0) in
    let r = ( (- sin phi) :+ k*ly )*((cos(theta - phi))/ly :+ 0.0) in i * r
  else if ( (yg==ny) && (xg<nx) ) then
    let i = (mkPolar lx (k*yc*sin(phi)))*(mkPolar 1.0 (k*xc*cos(phi)))*
      ((sinc (k*lx*(cos phi)/2.0)) :+ 0.0) in
    let r = ( (- sin phi) :+ (- k*ly) )*((cos(theta - phi))/ly :+ 0.0) in i * r
  else 0.0 :+ 0.0
```

tion. Partly building on Ömer's work, Bettelli et al.² proposed a language that is an extension of C++. The guarded command language qGCL of Sanders and Zuliani¹⁶ hints at a language for program specification.

The first quantum programming language in the style of functional programming was the quantum lambda calculus of Selinger and Valiron,¹⁷ providing a unified framework for manipulating classical and quantum data.

The quantum lambda calculus has a well-defined mathematical semantics that guarantees the absence of runtime errors in a well-typed program. The language is easily extended with inductive data types (such as lists) and recursion. One shortcoming of the quantum lambda calculus, however, is that it does not separate circuit construction from circuit evaluation. It thus lacks the ability to manipulate quantum circuits as data, as well as the ability to au-

Figure 17. The `calcRweights` circuit.



tomatically construct unitary circuits from a classical description. These problems were partly addressed by the Quantum IO Monad of Green and Altenkirch,⁷ a functional language that is a direct predecessor of Quipper.

The Quipper Language

Building on this previous work, we introduce Quipper, a functional language for quantum computation. We chose to implement Quipper as a deeply embedded domain-specific language (EDSL) inside the host language Haskell; see Gill⁶ for an overview of EDSLs and their embedding in Haskell. Quipper is intended to offer a unified general-purpose programming framework for quantum computation. Its main features are:

Hardware independence. Quipper's paradigm is to view quantum computation at the level of logical circuits. The addition of error-correcting codes and mapping to hardware are left to other components further down the chain of compilation;

Extended circuit model. The initialization and termination of qubits is explicitly tracked for the purpose of ancilla management;

Hierarchical circuits. Quipper features subroutines at the circuit level, or “boxed subcircuits,” permitting a compact representation of circuits in memory;

Versatile circuit description language. Quipper permits multiple programming styles and can handle both procedural and functional paradigms of computation. It also permits high-level manipulations of circuits with programmable operators;

Two runtimes. A Quipper program typically describes a family of circuits

that depends on some classical parameters. The first runtime is “circuit generation,” and the second runtime is “circuit execution.” In batch-mode operation, as discussed earlier, these two runtimes take place one after the other, whereas in online operation, they may be interleaved;

Parameter/input distinction. Quipper has two notions of classical data: “parameters,” which must be known at circuit-generation time, and “inputs,” which may be known only at circuit-execution time. For example, the type `Bool` is used for Boolean parameters, and the type `Bit` is used for Boolean inputs;

Extensible data types. Quipper offers an abstract, extensible view of quantum data using Haskell's powerful type-class mechanism; and

Automatic generation of quantum oracles. Many quantum algorithms require some nontrivial classical computation to be made reversible and then lifted to quantum operation. Quipper has facilities for turning an ordinary Haskell program into a reversible circuit. This feature is implemented using a self-referential Haskell feature known as “Template Haskell” that enables a Haskell program to inspect and manipulate its own source code.

Quipper Feature Highlights

We briefly highlight some of Quipper's features with code examples:

Procedural paradigm. In Quipper, qubits are held in variables, and gates are applied one at a time. The type of a circuit-producing function is distinguished by the keyword `Circ` after the arrow, as in Figure 7. The function `mycirc` inputs `a` and `b` of type `Qubit` and outputs a pair of qubits while gen-

erating a circuit;

Block structure. Functions generating circuits can be reused as subroutines to generate larger circuits. Operators (such as `with_control`) can take an entire block of code as an argument, as in Figure 8. Note “do” introduces an indented block of code, and “\$” is an idiosyncrasy of Haskell syntax that can be ignored by the reader here;

Circuit operators. Quipper can treat circuits as data and provide high-level operators for manipulating whole circuits. For example, the operator `reverse_simple` reverses a circuit, as in Figure 9;

Circuit transformers. Quipper provides user-programmable “circuit transformers” as a mechanism for modifying a circuit on a gate-by-gate basis. For example, the `timestep` circuit in Figure 9 can be decomposed into binary gates using the `Binary` transformer, as in Figure 10; and

Automated functional-to-reversible translation. Quipper provides a special keyword `build_circuit` for automatically synthesizing a circuit from an ordinary functional program, as in Figure 11. The resulting circuit can be made reversible with the operator `classical_to_reversible`, as in Figure 12.

Experience with Quipper

We have used Quipper to implement seven nontrivial quantum algorithms from the literature, based on documents provided by the Quantum Computer Science program of the U.S. Intelligence Advanced Research Projects Activity (IARPA).¹⁰ All of these algorithms can be run, in the sense that we can print the corresponding circuits for small parameters and perform automated gate counts for circuits of less tractable sizes. Each of these algorithms (see the table here) solves some problem believed to be classically hard, and each algorithm provides an asymptotic quantum speedup, though not necessarily an exponential one. These seven algorithms cover a broad spectrum of quantum techniques; for example, the table includes several algorithms that use the Quantum Fourier Transform, phase estimation, Trotterization, and amplitude amplification. IARPA selected the algorithms for being comparatively complex, which

A selection of quantum algorithms.

Algorithm	Description
Binary Welded Tree ⁴	Finds a labeled node in a graph by performing a quantum walk
Boolean Formula ¹	Evaluates an exponentially large Boolean formula using quantum simulation; QCS version computes a winning strategy for the game of Hex
Class Number ⁸	Approximates the class group of a real quadratic number field
Ground State Estimation ¹⁹	Computes the ground state energy level of a molecule
Quantum Linear Systems ⁹	Solves a very large but sparse system of linear equations
Unique Shortest Vector ¹⁵	Finds the shortest vector in an n -dimensional lattice
Triangle Finding ¹²	Finds the unique triangle inside a very large dense graph

is why some better-known but technically simpler algorithms (such as Shor's factorization algorithm) were not included.

Using Quipper, we are able to perform semi- or completely automated logical-gate-count estimations for each of the algorithms, even for very large problem sizes. For example, in the case of the triangle-finding algorithm, the command

```
./tf -f gatecount
-o orthodox
-n 15 -l 31 -r 6
```

produces the gate count for the complete algorithm on a graph of 2^{15} vertices using an oracle with 31-bit integers and a Hamming graph of tuple size 2^6 . This command runs to completion in less than two minutes on a laptop computer and produces a count of 30,189,977,982,990 gates and 4,676 qubits for this instance of the algorithm.

Examples. As a further illustration, here are two subroutines written in Quipper:

Procedural example. First, we formalize the circuit family in Figure 3. This circuit implements the time step for a quantum walk in the Binary Welded Tree algorithm.⁴ It inputs a list of pairs of qubits (a_i, b_i) , and a single qubit r . It first generates an ancilla, or scratch-space, qubit in state $|0\rangle$. It then applies the two-qubit gate W to each of the pairs (a_i, b_i) , followed by a series of doubly controlled NOT-gates acting on the ancilla. After a middle gate e^{izt} , it applies all the gates in reverse order. The ancilla ends up in the state $|0\rangle$ and is no longer needed. The Quipper code is in Figure 13, yielding the circuit in Figure 14. If one replaces the “3” with a “30” in the main function, one obtains a larger instance of this circuit family, as in Figure 15; and

A functional-to-reversible translation. This example is from the Quantum Linear Systems algorithm.⁹ Among other things, this algorithm contains an oracle calculating a vector r of complex numbers; Figure 16 shows the core function of the oracle. Note it relies heavily on algebraic and transcendental operations on real and complex numbers (such as $\sin$, $\cos$, sinc , and mkPolar), as well as on subroutines (such as edgetoxy and itoxoy) not shown in Figure 16. This function is

readily processed using Quipper's automated circuit-generation facilities. Algebraic and transcendental functions are mapped automatically to quantum versions provided by an existing Quipper library for fixed-point real and complex arithmetic. The result is the rather large circuit in Figure 17.

Conclusion

Practical quantum computation requires a tool chain extending from abstract algorithm descriptions down to the level of physical particles. Quantum programming languages are an important aspect of this tool chain. Ideally, such a language enables a quantum algorithm to be expressed at a high level of abstraction, similar to what may be found in a research paper, and translates it down to a logical circuit. We view this logical circuit as an intermediate representation that can then be further processed by other tools, adding quantum control and error correction, and finally passed to a real-time system controlling physical operations.

Quipper is an example of a language suited to a quantum coprocessor model. We demonstrated Quipper's feasibility by implementing several large-scale algorithms. The design of Quipper solved some major programming-language challenges associated with quantum computation, but there is still much to do, particularly in the design of type systems for quantum computing. As an embedded language, Quipper is confined to using Haskell's type system, providing many important safety guarantees. However, due to Haskell's lack of support for linear types, some safety properties (such as the absence of attempts to clone quantum information) are not adequately supported. The design of ever better type systems for quantum computing is the subject of ongoing research. 

References

1. Ambainis, A., Childs, A.M., Reichardt, B.W., Špalek, R., and Zhang, S. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM Journal on Computing* 39, 2 (2010), 2513–2530.
2. Bettelli, S., Calarco, T., and Serafini, L. Toward an architecture for quantum programming. *The European Physical Journal D* 25, 2 (2003), 181–200.
3. Burham, H., Durr, C., Heiligman, M., Hoyer, P., Magniez, F., Santha, M., and de Wolf, R. Quantum algorithms for element distinctness. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity* (Chicago, June 18–21). IEEE Computer Society Press, 2001, 131–137.

4. Childs, A. M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., and Spielman, D.A. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing* (San Diego, CA, June 9–11). ACM Press, New York, 2003, 59–68.
5. Gay, S.J. Quantum programming languages: Survey and bibliography. *Mathematical Structures in Computer Science* 16, 4 (2006), 581–600.
6. Gill, A. Domain-specific languages and code synthesis using Haskell. *Commun. ACM* 57, 6 (June 2014), 42–49.
7. Green, A. and Altenkirch, T. The quantum IO monad. In *Semantic Techniques in Quantum Computation*, S. Gay and I. Mackie, Eds. Cambridge University Press, Cambridge, U.K., 2009, 173–205.
8. Hallgren, S. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. *Journal of the ACM* 54, 1 (Mar. 2007), 4:1–4:19.
9. Harrow, A.W., Hassidim, A., and Lloyd, S. Quantum algorithm for solving linear systems of equations. *Physical Review Letters* 103, 15 (Oct. 2009), 150502-1–150502-4.
10. IARPA Quantum Computer Science Program. *Broad Agency Announcement IARPA-BAA-10-02*, Apr. 2010; <https://www.fbo.gov/notices/637e87ac1274d030ce2ab69339ccf93c>
11. Knill, E.H. *Conventions for Quantum Pseudocode*. Technical Report LAUR-96-2724. Los Alamos National Laboratory, Los Alamos, NM, 1996.
12. Magniez, F., Santha, M., and Szegedy, M. Quantum algorithms for the triangle problem. *SIAM Journal on Computing* 37, 2 (2007), 413–424.
13. Meter, R.V. and Horsman, C. A blueprint for building a quantum computer. *Commun. ACM* 56, 10 (Oct. 2013), 84–93.
14. Ömer, B. *Quantum Programming in QCL*. Master's Thesis. Institute of Information Systems, Technical University of Vienna, Vienna, Austria, 2000; tpu.tuwien.ac.at/~oemer/qcl.html
15. Regev, O. Quantum computation and lattice problems. *SIAM Journal on Computing* 33, 3 (2004), 738–760.
16. Sanders, J.W. and Zuliani, P. Quantum programming. In *Proceedings of the Fifth International Conference on Mathematics of Program Construction*, Vol. 1837 of *Lecture Notes in Computer Science* (Ponte de Lima, Portugal, July 3–5). Springer-Verlag, Berlin Heidelberg, 2000, 80–99.
17. Selinger, P. and Valiron, B. A lambda calculus for quantum computation with classical control. *Mathematical Structures in Computer Science* 16, 3 (2006), 527–552.
18. van Tonder, A. A lambda calculus for quantum computation. *SIAM Journal of Computation* 33, 5 (2004), 1109–1135.
19. Whitfield, J.D., Biamonte, J., and Aspuru-Guzik, A. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics* 109, 5 (Mar. 2011), 735–750.

Benoît Valiron (benoit.valiron@monoidal.net) is an assistant professor in the engineering school CentraleSupélec and a researcher in the Computer Science Laboratory of the Université Paris Sud, Paris, France.

Neil J. Ross (neil.j.ross@gmail.com) is a Ph.D. candidate at Dalhousie University, Halifax, Nova Scotia, Canada.

Peter Selinger (selinger@mathstat.dal.ca) is a professor of mathematics at Dalhousie University, Halifax, Nova Scotia, Canada.

D. Scott Alexander (salexander@apccomsci.com) is a chief scientist at Applied Communication Science, Basking Ridge, NJ.

Jonathan M. Smith (jms@cis.upenn.edu) is the Olga and Alberico Pompa Professor of Engineering and Applied Science and a professor of computer and information science at the University of Pennsylvania, Philadelphia, PA.

Copyright held by authors.

Publication rights licensed to ACM. \$15.00



Watch the authors discuss their work in this exclusive *Communications* video. <http://cacm.acm.org/videos/programming-the-quantum-future>

DOI:10.1145/2699410

Legitimacy of surveillance is crucial to safeguarding validity of OSINT data as a tool for law-enforcement agencies.

BY PETRA SASKIA BAYERL AND BABAK AKHGAR

Surveillance and Falsification Implications for Open Source Intelligence Investigations

OPEN SOURCE INTELLIGENCE, or OSINT, has become a permanent fixture in the private sector for assessing consumers' product perceptions, tracking public opinions, and measuring customer loyalty.¹² The public sector, and, here, particularly law-enforcement agencies, including police, also increasingly acknowledge the value of OSINT techniques for enhancing their investigative capabilities and response to criminal threats.⁵

OSINT refers to the collection of intelligence from information sources freely available in the public

domain, including offline sources (such as newspapers, magazines, radio, and television), along with information on the Internet.^{4,16,17} The spread of social media has vastly increased the quantity and accessibility of OSINT sources.^{3,11} OSINT thus complements traditional methods of intelligence gathering at very low to no cost.^{4,15}

OSINT increasingly supports the work of law-enforcement agencies in identifying criminals and their activities (such as recruitment, transfer of information and money, and coordination of illicit activities);¹⁸ for instance, the capture of Vito Roberto Palazzolo, a treasurer for the Italian mafia on the run for 30 years was accomplished in part by monitoring his Facebook account.⁸ OSINT also demonstrated its potential to help respond quickly to criminal behavior outside the Internet, as during, for instance, public disorder (such as the 2011 U.K. riots).¹ OSINT has therefore become an important tool for law-enforcement agencies combating crime and ultimately safeguarding our societies.¹⁴

To fulfill these functions, OSINT depends on the integrity and accuracy of open data sources. This integrity is jeopardized if Internet users choose not to disclose personal information or even provide false information on

» key insights

- **Falsification of personal information online is widespread, though users do not falsify information in a uniform way; some types of information are more likely to be falsified, leading to systematic differences in the reliability of the information for OSINT applications.**
- **Acceptance of and propensity for falsification is linked to attitudes toward online surveillance; the more negative people's attitudes toward governmental online surveillance, the more likely they are to accept information falsification and provide false information.**
- **The source of online surveillance—state agencies vs. private companies vs. unnamed organizations—seems to affect whether assumptions about online surveillance are linked to information falsification.**



themselves.^{7,9} Such omissions and falsifications can have grave consequences if decisions are being made from data assumed to be accurate but that is not.¹⁹

This issue is especially relevant since the revelations by former NSA contractor Edward Snowden of large-scale monitoring of communications and online data by state agencies

worldwide. The revelations created considerable mistrust by citizens of Internet-based surveillance by their own governments, bringing the tension between the security of society and the fundamental right to privacy into sharp profile. These discussions begin to show concrete effects; for instance, use of privacy-sensitive keywords in Google searches changed

from the period before to the period after Snowden's revelations, as users proved less willing to use keywords "that might get them in trouble with the [U.S.] government."¹⁰ Despite mandatory national and international data protection and privacy regulations, Internet users thus seem wary of online surveillance and in consequence modify their online behavior.

For organizations using OSINT in their decision making, changes in users' online behaviors, specifically their willingness to provide accurate accounts about themselves, are problematic. Not only do they increase the incidence of false information, they also raise the complexity and costs for information validation, or authentication of individuals' Web footprints, against additional and trusted sources.

A better understanding of the tendency of Internet users of when and why to change their online behavior in response to online surveillance can help pinpoint especially problematic areas for the validity of OSINT methods. Such an understanding can further guide efforts for more targeted cross validations. So far, organizations, including law-enforcement agencies, lack a clear

picture of how far and in what ways concerns about online surveillance change informationbases relevant to law-enforcement agencies' use of open source intelligence. In our current research, we aim to systematically investigate whether shifts in online behaviors are likely and, if so, what form they might take. In this article, we report on a study in which we focused on the falsification of personal information, investigating the link between falsification acceptance and falsification propensity with attitudes toward online surveillance, privacy concerns, and assumptions about online surveillance by different organizations.

Study Design and Sample

To understand Internet users' attitudes toward falsification of personal information in connection with online surveillance, we conducted an online survey between January and March 2014 using the micro-work platform Amazon Mechanical Turk to recruit participants.^a A total of 304 users responded to our request, of which 298 provided usable answers. Our sample consisted largely of experienced Internet users (72.2% had more than 11 years of experience) and intensive users (41.3% using the Internet for at least seven hours per day). The majority (83.9%) of participants lived in the U.S., 9.4% in India, and the others were from Canada, Croatia, Kenya, or Romania (0.4% to 1.1% per country). The gender distribution was nearly equal, with 48.9% male vs. 50.4% female participants; 0.7% preferred not to answer the question. Participants were relatively young, with a majority 40 years or younger (67.3%), of which most (35.6%) were between 21 and 30 years of age. Older participants were slightly underrepresented, with 9.5% between 51 and 60 and 3.9% over 60; 0.7% preferred not to answer the question. The questionnaire was administered online. On completion of the survey participants were paid \$0.70 through the Mechanical Turk platform. The survey took an average four minutes to complete.

^a Amazon Mechanical Turk is an online service that allows recruitment of participants worldwide for jobs of short duration, often lasting only several minutes; see <https://www.mturk.com>

Figure 1. Respondents' attitudes toward the positive and negative sides of state online surveillance.

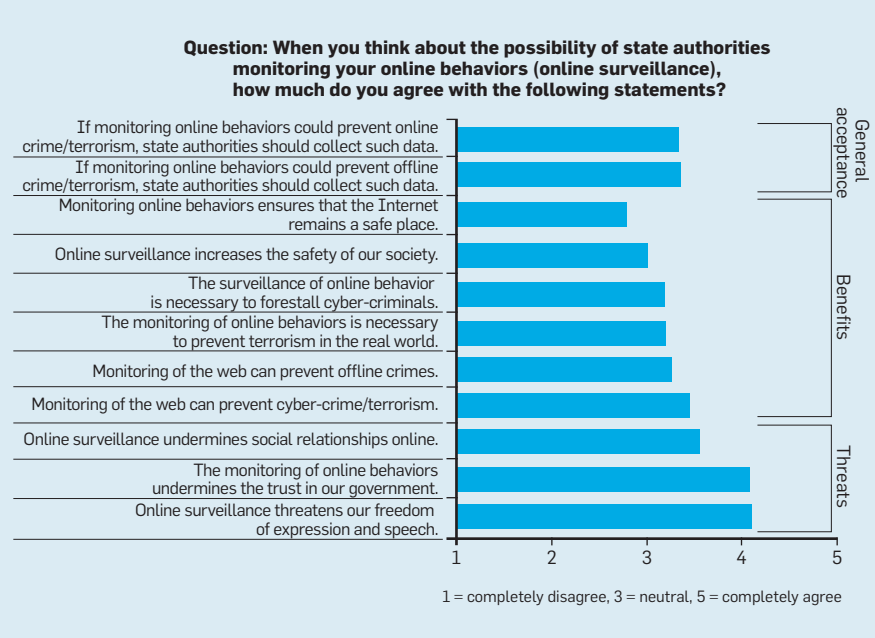
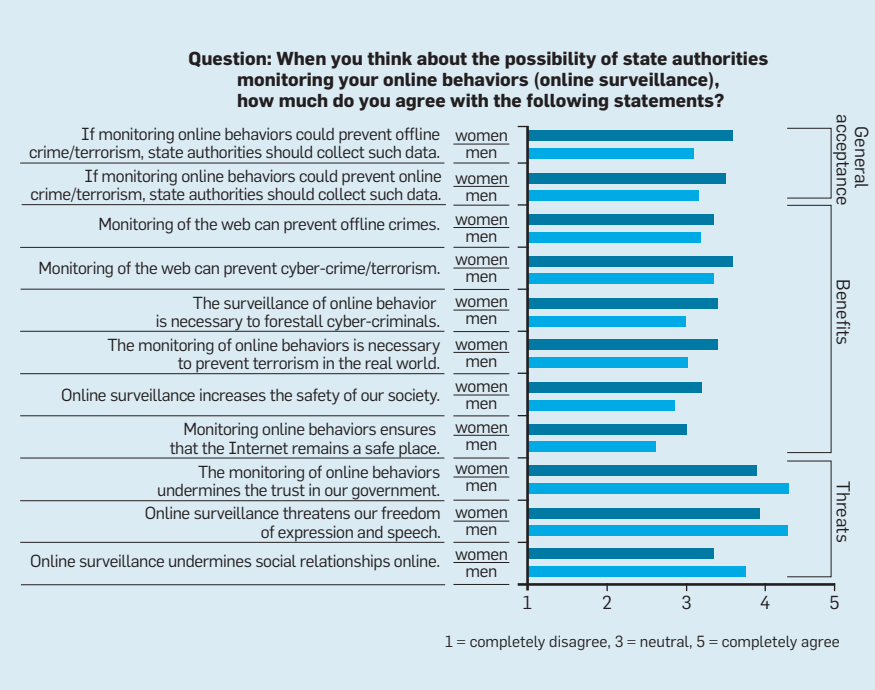


Figure 2. Gender differences in perceived benefits and threats of state online surveillance.



Findings

In the following sections, we detail our findings on participants' attitudes toward surveillance, acceptance of and propensity for falsification of their personal online information, and the possible links between them.

Attitudes toward online surveillance by state agencies. The first question when investigating the effect of state surveillance on online behaviors is how Internet users perceive its value. To capture attitudes toward online surveillance by state agencies we asked participants to indicate their agreement with 11 statements, five of them positive toward online surveillance, thus addressing potential benefits, three of them negative, addressing possible threats, and two capturing general acceptance; Figure 1 shows the average values for benefits, threats, and general acceptance for the entire sample.

The general acceptance of online surveillance was at a medium level with $m = 3.35$ when the focus was on the prevention of offline crimes and $m = 3.33$ when focusing on the prevention of online crimes (both measured on a scale of 1 to 5). Overall, negative attitudes were considerably stronger than positive attitudes. Participants were especially concerned about threats to freedom of expression and speech and the undermining of trust in their own government. Interestingly, the claims state agencies often make that monitoring online behavior ensures the Internet stays a safe place or increases the safety of society found little agreement.

Women were generally more accepting of online surveillance ($t(280) = -3.02$, $p < .01$), seeing significantly more benefits than men ($t(279) = -2.60$, $p < .01$). Men in contrast reported significantly more concern about its negative aspects ($t(275) = 3.69$, $p < .001$) (see Figure 2). Women were especially more willing to support online surveillance if it could prevent crimes perpetrated outside the Internet (offline crimes), whereas men were particularly concerned about the undermining of trust in the government. Moreover, users with more experience in the use of the Internet (more than 11 years) were significantly less positive toward online surveillance than users with less ex-

Figure 3. Respondents' assumptions concerning the degree of online surveillance by different organizations.

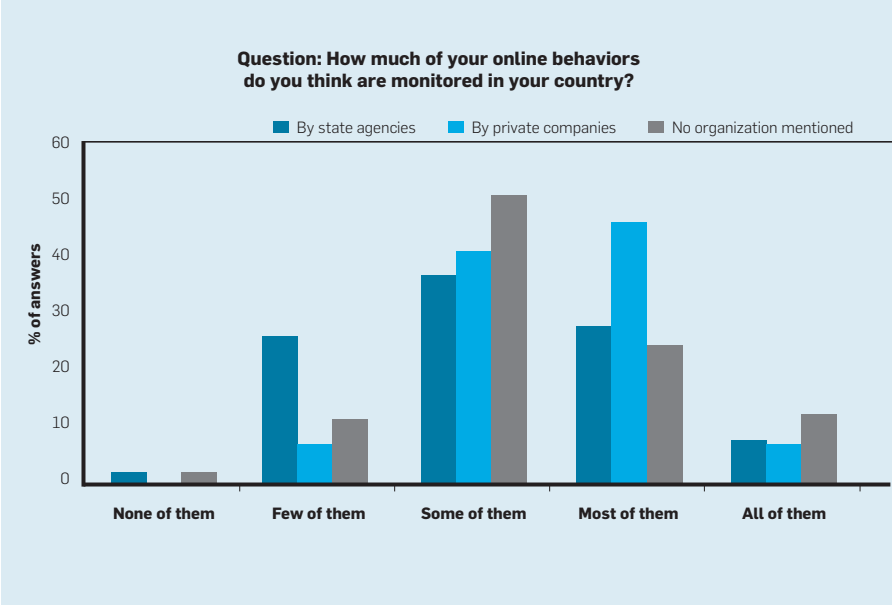
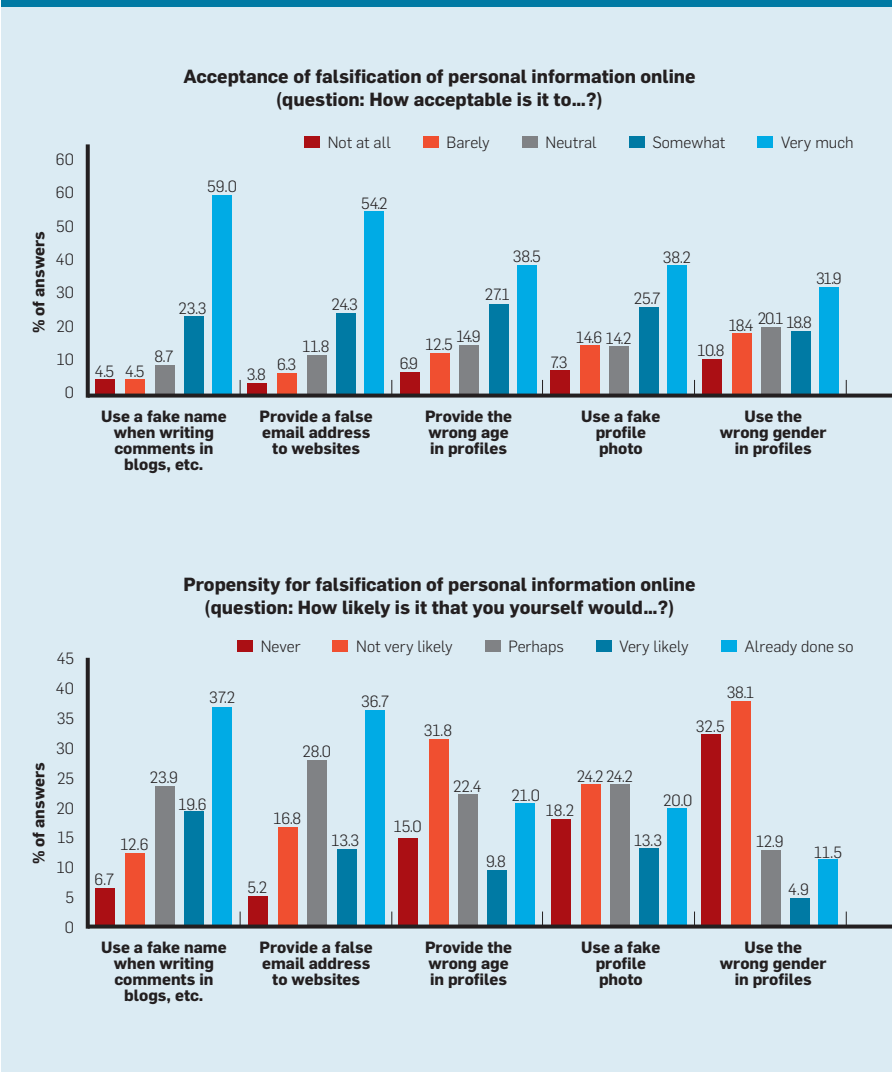


Figure 4. Acceptance and propensity for falsification of personal information among all participants.



perience (seven years or less; $F(2,274) = 5.04, p < .01$). Since age groups did not differ in their attitudes, this effect cannot be explained by generational differences. It instead hints at growing sensitivity toward the issue with increased Internet use.

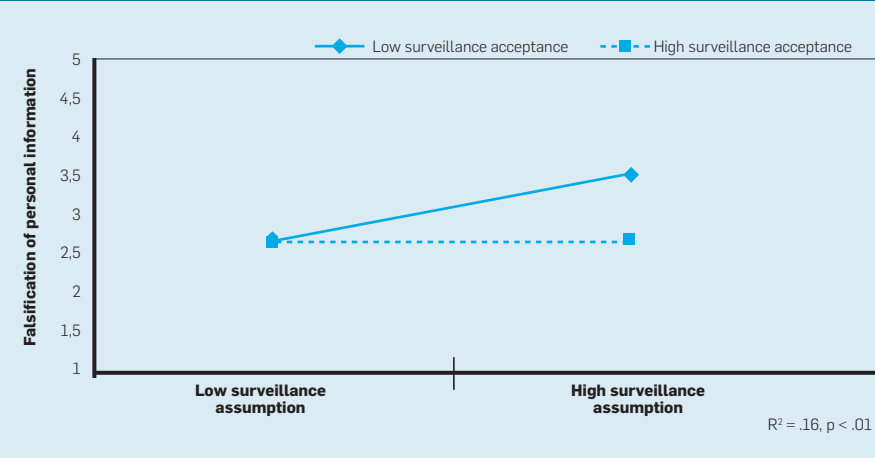
Surveillance by state agencies vs. private companies or unnamed organizations. Unlike private companies, which are widely known for collecting online data on a large scale, OSINT use by state agencies has only recently come to the attention of the broader public. Yet, as demonstrated by the intense discussion in the aftermath of the Snowden

revelations, the sensitivity of the issue seems even greater. Also, compared to the use of OSINT by private companies, the consequences of OSINT use by law-enforcement agencies can be considerably more severe for the individual under scrutiny. We therefore wanted to know whether online surveillance by state agencies could lead to different reactions from surveillance conducted by private industry. For the second part of the survey we used three different framings for our questions, one mentioning surveillance conducted by state agencies, one mentioning surveillance by private companies, and one men-

tioning surveillance without naming a specific organization. A total of 104 people, or 34.9% of respondents, filled out the survey referring to surveillance by state authorities; 103 people (34.6%) answered the survey referring to surveillance by private companies; and 91, or 30.5%, reacted to the generalized condition in which no specific organization was named.

First, we were interested in the extent of online surveillance users assumed across the three sources of surveillance, ranging from “none” of their online behaviors to “all of them.” In all three conditions, the average values indicate users assumed at least some of their behavior is monitored, although the values were highest for private companies ($m = 3.52$) and lowest for state agencies ($m = 3.13$) (see Figure 3). This difference was also statistically significant ($F(2,294) = 5.37, p < .01$). This was a general tendency, as genders, age groups, and user groups with different degrees of Internet experience did not differ in their assumptions about online surveillance. Despite current debates, private companies thus seem to be perceived as more intrusive than state agencies. As we outline in the following sections, this does not mean, however, that surveillance by state

Figure 5. Role of surveillance assumptions and acceptance in information falsification.



Correlations between falsification behaviors and online surveillance assumptions and attitudes.

Generic condition (no mention of an organization; n = 91)

	Mean	Std. dev.	1.	2.
1. Assumption of online surveillance	3.36	0.88		
2. Acceptance of information falsification	3.80	1.06	.22	
3. Propensity for information falsification	3.02	1.03	.10	.66**

Condition “surveillance by private companies” (n = 103)

	Mean	Std. dev.	1.	2.
1. Assumption of online surveillance	3.52	0.73		
2. Acceptance of information falsification	3.99	0.96	.13	
3. Propensity for information falsification	3.26	1.03	.12	.63**

Condition “surveillance by state agencies” (n = 104)

	Mean	Std. dev.	1.	2.	3.	4.	5.
1. Assumption of online surveillance	3.13	0.96					
2. General acceptance of online surveillance by state agencies	3.23	1.22	-.04				
3. Benefits from surveillance	3.06	1.02	.01	.78**			
4. Threats from surveillance	4.05	0.79	.11	-.38**	-.49**		
5. Acceptance of information falsification	3.84	0.96	.08	-.32**	-.24**	.21*	
6. Propensity for information falsification	2.92	1.07	.24*	-.26**	-.23*	.13	.59**

* $p < .05$

** $p < .01$; Pearson correlations, two-sided tests

agencies is seen as less intrusive than that of private companies.

Degree of acceptance and propensity to falsify personal information online. To understand whether concerns about online surveillance affect the tendency to falsify personal information online, we asked participants in all three conditions the same two questions: How acceptable is it to falsify personal online information (acceptance of falsification, from 1 = not at all to 5 = very much)? And how likely are you to falsify your own personal information online (propensity for falsification, from 1 = never to 5 = already done so)?

We asked participants about the falsification of five types of information that are fixtures in most online profiles: providing a false name, providing a fake email address, providing the wrong age, using a fake photo, and providing the wrong gender.

Taking all five together, users showed a high level of acceptance for falsification ($m = 3.88$, $SD = 0.99$), while the propensity for falsification was somewhat less ($m = 3.06$, $SD = 1.05$). Still, only a very small group (3.4%) indicated they would never falsify any information, whereas 7.4% indicated having already done so for all five categories.

Interestingly, falsification acceptance and propensity were not uniform across all five. Using a false name and false email address was seen as acceptable, whereas a false profile photo and wrong gender were considered much less acceptable (see top part of Figure 4). Only 9.0% of the participants considered falsifying their own name as completely or highly unacceptable; for the falsification of one's gender, this was 29.2%. The same trend emerged for the propensity of falsifying information; 37.0% of participants indicated they had already used a fake name and email address, while 70.6% reported they would never use the wrong gender or were very unlikely to do so (see bottom part of Figure 4). Users thus seem nearly five times more likely to indicate the wrong name and more than six times more likely to provide a wrong email address than report the wrong gender. This suggests the falsification of personal information follows specific patterns; that is, dif-



The more participants perceived online surveillance by state agencies as problematic, the more willing they were to accept falsification.



ferent pieces of information in a profile may have a disparate likelihood of being valid or invalid, or “differential validity of information types.”

To compare the effect of the three surveillance sources, we summarized the five types of information into one score for acceptance and one score for propensity, respectively. The three conditions did not differ in terms of falsification acceptance ($F(2,285) = 0.92$, nonsignificant) but resulted in at least a marginal effect for falsification propensity ($F(2,281) = 2.77$, $p = .06$). This was due to a slightly greater propensity for falsification when surveillance was conducted by private companies ($m = 3.26$) compared to state agencies ($m = 2.91$; $t = -2.29$, $p < .05$). Gender, age groups, and length of Internet use had no effect on either outcome.

Linking information falsification with surveillance assumptions and attitudes. We next considered influence of surveillance awareness, attitudes toward surveillance, and privacy concerns on information falsification. Because we used three separate versions of the survey to determine the influence of the organization conducting surveillance, the questions on degree of surveillance awareness and falsification acceptance and propensity referred to different entities: state agencies, private organizations, or no organization in particular. We therefore calculated the correlations between surveillance awareness and information falsification for each of the three groups separately. This also gave us the opportunity to investigate whether the context of surveillance had an effect on falsification behaviors. The table here reports the results for each of the three conditions.

Interestingly, assumptions of online surveillance had an effect on falsification acceptance and propensity only when online surveillance was framed in the context of state agencies or as generalized activity. In these cases, assumptions about online surveillance had a clear positive link with either the propensity to falsify personal information or the acceptance of this behavior, as in the table. For surveillance conducted by private companies, no such significant link emerged. Again, this suggests the question of who conducts the sur-

veillance may play a role in influencing concrete falsification behaviors. Surveillance by state agencies could trigger more concrete reactions than either generalized surveillance or monitoring by private companies.

As in the third generalized condition, all questions referred uniformly to state agencies, this subgroup of participants gave us the opportunity to further investigate the link between attitudes toward online surveillance by state agencies and falsification. In this subgroup, we found a clear link between attitudes toward online surveillance, acceptance, and propensity for falsification. The greater the general acceptance and perceived benefits of surveillance, the less accepting participants were of falsifying information and the less likely they were to do it themselves. Similarly, the more participants perceived online surveillance by state agencies as problematic, the more willing they were to accept falsification.

In addition, acceptance of online surveillance moderated the relationship between falsification and assumed degree of surveillance. While greater assumptions of surveillance generally increased the propensity for falsification, this reaction was especially strong for people with a low acceptance of online surveillance by state agencies (see Figure 5). This observation suggests an important interaction between awareness and attitudes. While surveillance awareness alone may lead to information falsification, the main trigger to falsifying personal information seems to be the extent surveillance is seen as (in)appropriate. This logic links tendencies for falsification of one's own information to how much one considers state agencies legitimate and trustworthy, thus emphasizing the potentially critical effect of negative press on the viability of OSINT-based decisions.

More than a Moral Dilemma

Our study demonstrates that discussions about “privacy” vs. the “rightfulness” of online surveillance is more than a moral dilemma. Rather, the degree to which individuals are aware of online surveillance and the way they view the acceptability of this act, including the organizations implicated



Law-enforcement agencies will have to become more sensitive to the reactions their own practices might create for the viability of their methods and in consequence the decisions they take based on these methods.



in it, can pose concrete challenges for the validity of online data—and consequently for the validity of decisions based on the data. While our study is only a small window into this complex issue, it demonstrates that online surveillance may have very concrete, practical implications for the use and usefulness of OSINT, specifically for law-enforcement agencies. Surveillance is not neutral. On the contrary, our study attests that surveillance practices could threaten the integrity of the very data they rely on.

Falsification tendencies as a reaction to online surveillance create challenges for the usability of open source data, especially increasingly for the effort required to validate information. OSINT has long been hailed as a cheap or even “no cost” source of operational information for law-enforcement agencies.^{4,16} Our findings suggest that increasing awareness of online surveillance, including painful revelations of problematic surveillance practices by states and law-enforcement agencies, may severely reduce this benefit, at least for those Internet users with a more critical outlook toward state authorities and/or greater need for privacy.

Technical solutions to counter the increased likelihood of falsification are available; for instance, Dai et al.⁵ proposed a number of “trust score” computation models that try to determine data trustworthiness in anonymized social networks using a trusted standard. Additional solutions are thinkable using validity pattern mining, reasoning-based semantic data mining, and open source analysis techniques. One important avenue for identifying false information is to identify possible links between profiles of a single user and then mine the data between profiles for validation. Users often explicitly link their profiles. For example, Twitter posts and Instagram photos can be organized so they appear on a user's Facebook timeline. This gives a direct and verified link to further information. Users may also post under the same pseudonym on a number of profiles. Collecting the data associated with each of these profiles provides further opportunity for corroboration.

As with Dai et al.,⁵ another tactic might be to attempt to match the social graph of users across networks. Inconsistencies in personal data may be identified by verifying where these networks overlap.

The most difficult part in information validations is determining the technological solutions that must be employed to carry out the validation. Two such techniques are classification and association mining. Machine-learning-based classification techniques can be used to establish a ground-truth dataset containing information known to be accurate. By training models on this data, outliers in new data could indicate the trustworthiness of the information may warrant further investigation. Association mining (or association rule learning) can be used to discover relationships between variables within datasets, including social media and other OSINT sources.¹² These association rules can take data from the links discovered between multiple social networks and be used to validate the existing information.

Still, all these technical solutions rely on the cross-validation of open source information with other (open or closed) sources. Growing falsification tendencies in the wake of increasing online surveillance awareness will make such cross-validations not only increasingly necessary but also more complex and costly. Here, the notion of differential validity, as evidenced in our data, may provide a valuable perspective toward a more systematic and targeted approach to information validation by guiding validation efforts toward more or less problematic data. This approach follows the observation that personal information seems to possess systematic variations in its veracity, leading to differential validity patterns. While our study focused on only a small set of static personal information, we assume similar patterns are also observable for other areas, as well as for more dynamic data.

An interesting question in this regard is how “volatile” falsifications of personal information tend to be. Do users stick with one type of falsification (such as consistently modifying

name, relationship status, or age) across services, or do these pieces of information vary across services? Also, do users always use the same content (such as the same false date of birth or photo)? Extending our knowledge of such falsification or validity patterns can considerably reduce the effort involved in validating OSINT-based data. In our current study, we did not investigate the reasons behind the differences in falsification acceptance and propensity for the various types of personal information. Getting a clearer understanding of these reasons could tell us much about the contexts in which falsification are more or less likely, as well as the strategies Internet users employ to remain private.

Conclusion

We clearly cannot return to the days of the “uninformed” or “unaware” Internet user, and law-enforcement agencies therefore need to find ways to deal with the consequences of online surveillance awareness by the general public and the possible ramifications it may have for the trustworthiness of online information. While we do not suggest OSINT will lose its value for investigation processes, we certainly think law-enforcement agencies will have to become more sensitive to the reactions their own practices might create for the viability of their methods and in consequence the decisions they take based on these methods.

Employing ever more advanced technical solutions is not the (sole) solution. Our findings make clear that even more than the pure fact of online surveillance, it is the perceived purpose and legitimacy of the act that are the main drivers behind the extent to which users alter their behaviors online. This explains the role of (largely negatively tinted) public discussions for the behavioral changes in the wake of Snowden’s revelations.¹⁰ They also highlight the criticality of properly legitimizing online surveillance to reduce distrust in law-enforcement agencies and thus pressures toward information falsifications and probably changes in online behaviors more generally. 

References

1. Bartlett, J., Miller, C., Crump, J., and Middleton, L. *Policing in an Information Age*. Demos, London, U.K., Mar. 2013.
2. Bell, P. and Congram, M. Intelligence-led policing (ILP) as a strategic planning resource in the fight against transnational organized crime (TOC). *International Journal of Business and Commerce* 2, 12 (2013), 15–28.
3. Best, C. Challenges in open source intelligence. In *Proceedings of the Intelligence and Security Informatics Conference* (Athens, Greece, Sept. 12–14, 2011), 58–62.
4. Best Jr., R.A. and Cumming, A. *Open Source Intelligence (OSINT): Issues for Congress*. Congressional Research Service, Washington, D.C., Dec. 2007; <https://www.fas.org/sgp/crs/intel/RL34270.pdf>
5. Dai, C., Rao, F.Y., Truta, T.M., and Bertino, E. Privacy-preserving assessment of social network data trustworthiness. In *Proceedings of the Eighth International Conference on Collaborative Computing* (Pittsburgh, PA, Oct. 14–17, 2012), 97–106.
6. Gibson, S. Open source intelligence: An intelligence lifeline. *The RUSI Journal* 149, 1 (2004), 16–22.
7. Joinson, A.N., Reips, U.D., Buchanan, T., and Schofield, C.B.P. Privacy, trust, and self-disclosure online. *Human-Computer Interaction* 25, 1 (2010), 1–24.
8. La Stampa. Mafia, fermato Vito Roberto Palazzolo scovato a Bangkok grazie a Facebook. *La Stampa* (Mar. 31, 2012); <http://www.lastampa.it/italia/cronache/mafia-fermato-vito-roberto-palazzolo-scovato-a-bangkok-grazie-a-facebook-vpnxhM5z5chH3iuIjttksJ/pagina.html>
9. Lenhart, A., Madden, M., Cortesi, S., Duggan, M., Smith, A., and Beaton, M. *Teens, Social Media, and Privacy*. Pew Internet and American Life Project Report, Washington, D.C., 2013; <http://www.pewinternet.org/2013/05/21/teens-social-media-and-privacy/>
10. Marthew, A. and Tucker, C. *Government Surveillance and Internet Search Behavior*. Working Paper. Social Science Research Network, Rochester, NY, Mar. 2014; https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2412564
11. Mercado, S.C. Sailing the sea of OSINT in the information age. *Studies in Intelligence* 48, 3 (2009), 45–55.
12. Nancy, P., Ramani, R.G., and Jacob, S.G. Mining of association patterns in social network data (Facebook 100 universities) through data mining techniques and methods. In *Proceedings of the Second International Conference on Advances in Computing and Information Technology*. Springer, Berlin, 2013, 107–117.
13. Neri, F., Aliprandi, C., Capecci, F., Cuadros, M., and By, T. Sentiment analysis on social media. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining* (Istanbul, Turkey, Aug. 26–29, 2012), 919–926.
14. Omand, D., Bartlett, J., and Miller, C. Introducing social media intelligence. *Intelligence and National Security* 27, 6 (2012), 801–823.
15. Ratzel, M.P. Europol in the combat of international terrorism. *NATO Security Through Science Series, Volume 19*. IOS Press, Amsterdam, 2007, 11–16.
16. Steele, R.D. The importance of open source intelligence to the military. *International Journal of Intelligence and Counter Intelligence* 8, 4 (1995), 457–470.
17. Steele, R.D. Open source intelligence. Chapter 10 in *Handbook of Intelligence Studies*. J. Loch, Ed. Routledge, New York, 2007, 129–147.
18. Stohl, M. Cyberterrorism: A clear and present danger, the sum of all fears, breaking point, or patriot games? *Crime, Law, and Social Change* 46, 4–5 (2006), 223–238.
19. The Telegraph. Connecticut school shooting: Police warn of social media ‘misinformation.’ *The Telegraph* (Dec. 16, 2012); <http://www.telegraph.co.uk/telegraph/tv/9748745/Connecticut-school-shooting-police-warn-of-social-media-misinformation.html>

Petra Saskia Bayerl (pbayerl@rsm.nl) is an assistant professor for technology and organizational behavior and program director technology of the Center of Excellence in Public Safety Management at the Rotterdam School of Management, Erasmus University Rotterdam, the Netherlands.

Babak Akhgar (B.Akhgar@shu.ac.uk) is a professor of informatics and director of the Center of Excellence in Terrorism, Resilience, Intelligence, and Organized Crime Research at Sheffield Hallam University, U.K.

DOI:10.1145/2716309

The National Palace Museum in Taiwan had to partner with experienced cloud providers to deliver television-quality exhibits.

BY RUA-HUAN TSAIH, DAVID C. YEN, AND YU-CHIEN CHANG

Challenges Deploying Complex Technologies in a Traditional Organization

AS MUSEUMS FOCUS increasingly on the public, they must develop new means of attracting and entertaining their visitors. Information and communication technologies (ICT) have great potential in this area. But deploying complex ICT in a traditional organizational setting like a museum is likely to be challenging.

The word “museum” comes from the Greek word “mouseion,” signifying both the seat of the Muses and a building specifically used to store and exhibit historic and natural objects. From a knowledge-management perspective, museums preserve, create, and share knowledge. In a museum, ancient wisdom

is preserved in objects and rediscovered through research. In terms of their intellectual contribution, museums create knowledge by recruiting researchers and giving them necessary incentives and resources. As a result, museums offer abundant knowledge to all. In the past, professional researchers worked independently to enhance their own expertise in their respective academic fields rather than for the benefit of the general public. Although museums stage exhibitions, visitors rarely have an intimate view of the related objects, which must be protected and preserved.

In line with social development, the function of museums has gradually changed over the years—first object-oriented (before 1980), then education-focused (1980s to 2000s), and finally public-centered (after 2000).⁴ The International Council of Museums⁹ redefined the 21st-century museum as follows: “A nonprofit institution in the service of society and its development, open to the public, which acquires, conserves, researches, communicates, and exhibits the tangible and intangible heritage of humanity and its environment for the purposes of education, study and enjoyment.”

Advanced information technologies (such as cloud computing) and communication technologies (such as global telephone systems and third- and fourth-generation mobile telecommunications technologies) are converging.

» key insights

- We developed a conceptual framework for how an organization can provide a new ICT-enabled service through a value-networkwide solution for establishing a service ecosystem.
- Any traditional organization must understand the needs of its business partners to be able to set up such an ecosystem.
- To implement a new value network for providing an ICT-enabled service, museums must consider these non-technical but ICT-related issues before and during development of projects related to a new service.



Adventures in the NPM: Poster for Formosa Odyssey.

This convergence has created a considerably less expensive ICT infrastructure that more effectively connects components, including information, knowledge, content, people, organizations, information systems, and other heterogeneous devices. As a result, new ICT-enabled services are available at lower cost to customers in general and members of the younger generation in particular. ICT-enabled services provide online, real-time interactive opportunities through applications; for example, a number of ICT-enabled services are offered through mobile applications.

Previous studies^{1,7,12} showed museums use computing technologies primarily to increase interactivity and enhance their visitors' experience visiting a museum. Various ICT-enabled museum services have been designed to meet the needs of the public; for example, the world's top museums, including the Louvre Museum in Paris, the Metropolitan Museum of Art in New York, and the National Palace Museum (NPM) in Taiwan, have established Facebook fan pages and other interactive, social, informative, entertaining online elements to stimulate awareness of and interest in their collections and encourage users to visit

their physical sites. Another example is the British Museum Channel playing video clips of exhibitions, collections, and behind-the-scenes experiences at the museum. This content is available on the museum's website. These technologies and platforms are expanding ways visitors access and interpret the objects displayed in museums.²

ICT-supported services enable museums to expand their social role and values by improving the timing and increasing the scope of their individual and collective engagement with visitors worldwide. Their services can be used to reach out to frequent museum visitors and potential visitors and non-visitors alike. Their purpose is to enhance the interaction and visiting experience of a broader range of visitors, on-site or online. Many such services are designed specifically to attract young people, who are often interested in experiencing new ICT and accustomed to using multimedia services in their daily lives.

Digital Archives Project

NPM was recognized in 2014 as one of the most visited museums in the world by *The Art Newspaper* and is distinguished by its extensive collection

of high-quality artifacts from Chinese history, making it one of the most popular destinations in Taiwan for international tourists. NPM management is supervised by the Executive Yuan, the highest administrative organ in the government of Taiwan.

NPM implemented the National Digital Archives Program (2002–2013) to digitize its collections on an ongoing basis, resulting in a large volume of high-quality IT-generated content on the museum's cultural and historical artifacts. In addition, NPM recognizes the novel opportunity provided by video and interactive objects to introduce historical treasures and revive interest in ancient artifacts. NPM also uses corporate advertising to stimulate the public's imagination and help them more fully appreciate China's historical artifacts. It continues to produce a number of videos related to collections, IT-generated content, and behind-the-scenes experience at the museum.

The videos produced by the museum have been extremely successful at sharing NPM's digital collection worldwide. In 2007, NPM used cutting-edge techniques to create a 3D animation called *Adventures in the NPM* that was

Figure 1. Components of the iPalace service system (adapted from Huang et al.⁸).

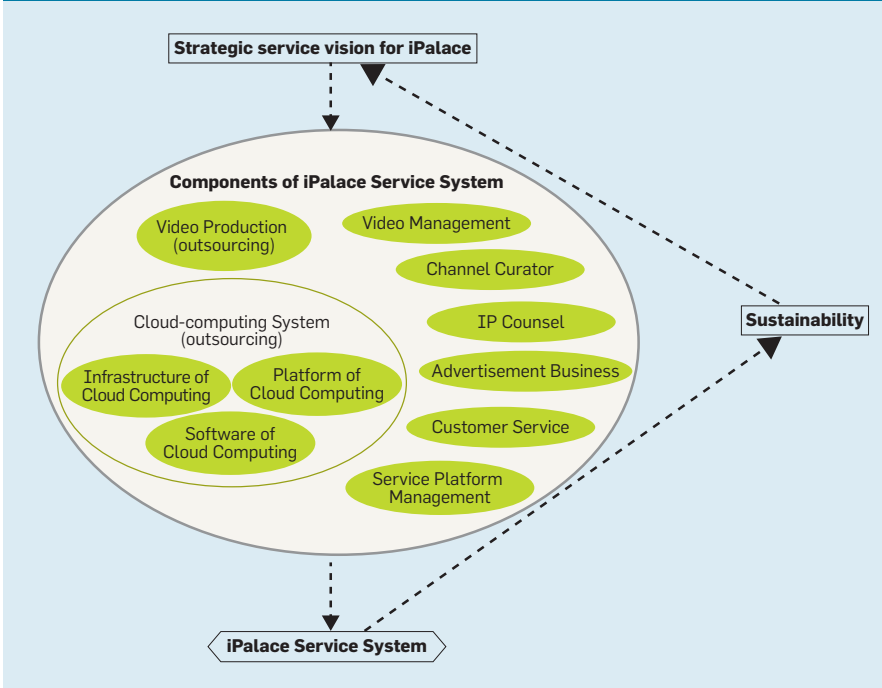


Figure 2. NIST cloud computing reference architecture (source: Liu et al.¹⁰).

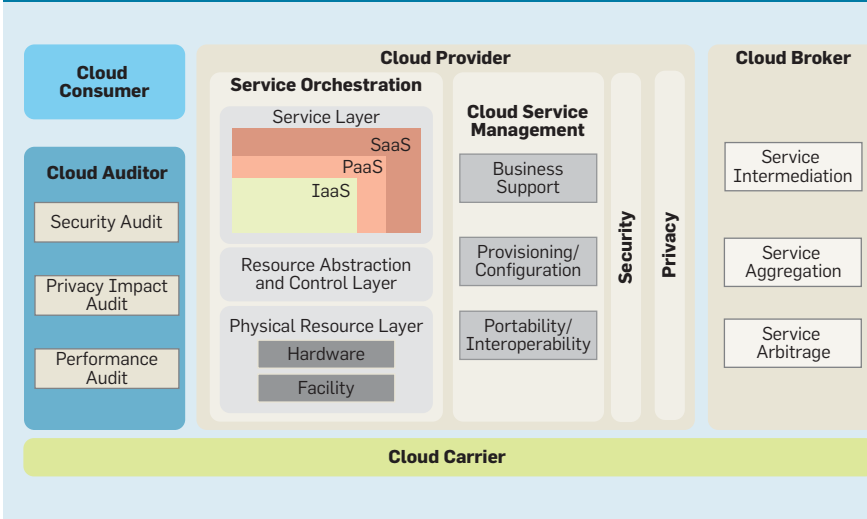
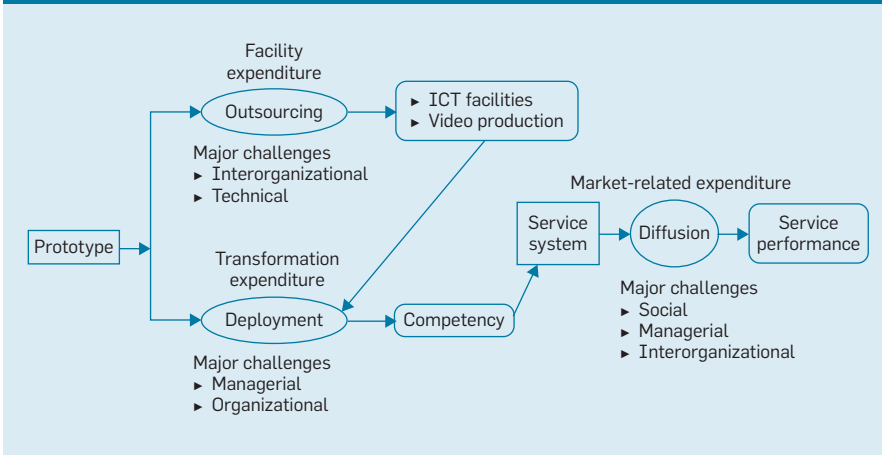


Figure 3. Process of diffusing iPalace from its prototype.



considered a milestone accomplishment, winning first prize in the public section of the 2008 Tokyo International Anime Fair, in addition to the Prix Coup de Coeur award at the in 2008 Festival International de l'Audiovisuel et du Multimédia sur le Patrimoine. At the 2009 Muse Awards, organized by the Media and Technology Professional Network of the American Alliance of Museums, NPM received a Silver Award for marketing development for a documentary called *Inside: The Emperor's Treasure* and a multimedia installation called *Passé-Future: The Future Museum of NPM*. In 2013, at the 46th Houston International Film Festival, NPM gained additional recognition by winning six major awards: two platinum, two gold, one bronze, and one special-jury. Its lighthearted comedic entry *Journeying from Past to Present—APP Minifilm* received a Platinum Award, the festival's highest accolade, in the network category. NPM plans to showcase its most representative collections on the global stage by internationally releasing its animated video *Adventures in the NPM*. Released in 2011, *Adventures in the NPM 2* featured treasures from the Palace Museum in Beijing to inspire and promote collaboration.

In 2012, NPM partnered with Google to display exquisite Chinese artifacts worldwide on Google's Art Project webpage. The Art Project initiative allowed NPM to publicly display its collections on an online platform, overcoming the temporal and spatial boundaries separating the museum from the rest of the world. NPM chose 18 artifacts familiar to the Taiwanese public to reach audiences worldwide on the Art Project webpage.

The experience collaborating with Google and creating so many high-quality videos inspired NPM to construct a video-streaming website to enable young people to access the museum's collections of inspiring Chinese artifacts. The iPalace initiative (<http://npm.nchc.org.tw>) was developed, then revised in December 2014 to address this goal.

iPalace

The table here outlines NPM strategic vision for the iPalace initiative. The target audience includes mainly young people who use Web browsers, are in-

interested in China's heritage, and enjoy videos and animations. The service concept emphasizes efficiently curated, well-organized video exhibitions that deliver fresh, attractive video content with smooth streaming in the form of a television program, giving viewers a high-quality online experience with NPM artifacts. Achieving these goals involves several operating strategies: a video-production process to ensure videos and animations are original and attractive; regular updating of the appearance of the interface to make it user friendly; and load balancing with such features as a task-oriented process design, an elastic Web service infrastructure, and peer-to-peer networking capability. As NPM's online counterpart, iPalace must deliver services that complement the museum's brand reputation while also coping with potential huge peaks in demand.

From the museum's perspective, iPalace was a radical innovation in ICT-enabled service, or RIIS. In addition, it is primarily a video-streaming service based on Web technology that provides a wonderful online visiting experience not linked to an in-gallery experience. Technically, however, iPalace requires expertise in sophisticated cloud-computing technology NPM does not have. From the public's perspective, iPalace is radically different from NPM's traditionally text-heavy webpages. Typical museum video channels offer a variety of video clips; for instance, the video content of the British Museum Channel is accessed through individual clicks on video clips. In contrast, operation of the iPalace initiative would be like a television program that broadcasts continuously until the viewer turns off the channel.

iPalace Value Network

The iPalace prototype was built in accordance with the strategic service vision within a reasonable timeframe and with limited effort. As the prototype was positively evaluated,³ NPM sought to transform the pilot service system into a real-world service system by implementing a full-scale value network. However, as outlined in Figure 1, the full iPalace value network includes video-production and cloud-computing services—areas where NPM lacked expertise. As a result, time and

effort beyond the museum's capability would have been necessary to vertically integrate the full value network. NPM thus chose a value-networkwide solution involving outsourcing video production and cloud computing. As discussed earlier, though NPM previously outsourced video production, it lacked experience outsourcing cloud-computing systems.

After several rounds of negotiation with potential outsourcing partners regarding the cloud-computing system, NPM became more realistic about how it could address the complexity of the iPalace value network; for instance, it recognized the usefulness of the cloud-computing reference architecture developed by the U.S. National Institute of Standards and Technology involving cloud consumers, cloud providers, cloud carriers, cloud auditors, and cloud brokers, as outlined in Figure 2. In outsourcing the cloud-computing system for iPalace, NPM functioned as a cloud customer aligned with other cloud actors identified as reliable strategic partners. Ultimately, it was necessary for NPM to collaborate with four categories of business partners in the iPalace value network:

Content makers. Studios, channel managers, and cloud operators;

Connectivity makers. Cloud operators, Internet service providers, and telephone operators;

Technology makers. Infrastructure manufacturers and middleware manufacturers; and

Sponsors. Agencies and advertisers providing sponsorship.

Leading the Value Network

Fitzsimmons and Fitzsimmons⁵ said one of the challenges facing all service innovators is how to achieve the required degree of integration; Figure 3 outlines the process of diffusing iPalace, beginning with its prototype. Related activities can be classified into three architectural categories—outsourcing, deployment, and diffusion—based on expected outcomes and underlying expenditure. Outsourcing activities involve expenditure on appropriate ICT-enabled facilities through interorganizational collaboration. Deployment activities require expenditure on transformation, as NPM and its partners must ensure the museum's existing capabilities and ICT facilities are able to support iPalace. Moreover, the museum's diffusion activities must be able to translate the new service system into concrete service performance through market-related expenditure on specific means of diffusing iPalace.

The challenges in the deployment process, as in Figure 3, relate to intra-organizational integration (such as employee acceptance involving employee culture and incentives and other organizational matters); processes in organizational management/reengineering (such as enforcement of interdepartmental collaboration or establishment of new departments and functions); and acceptance of a reference group. Additional professionals (such as curators

Strategic service vision for iPalace (adapted from Huang et al.⁸).

Service-Delivery System	Operating Strategy	Service Concept	Targeted Market Segments
Fresh, attractive video content	Continual provision of new video exhibitions	Video service that smoothly displays NPM relics:	Young people who ▶ Use Web browsers; ▶ Are interested in China's heritage; and ▶ Enjoy videos and animations.
Well-defined NPM experiences: ▶ NPM images; ▶ Ease of use; and ▶ In-depth insight into Chinese culture and aesthetics.	Periodically changing interface appearance	▶ Well-arranged exhibition; ▶ Available anytime, anywhere; and ▶ Uninterrupted multicasting.	
Smooth video delivery	Effective load balancing: ▶ Task-oriented process design; ▶ Elastic Web-service infrastructure; ▶ Peer-to-peer networking; and ▶ Distributed system architecture capable of multitasking.	Easy-to-use user interface: ▶ Updated for the occasion; and ▶ In-depth insight into Chinese culture and aesthetics.	

and marketers) must also be recruited to implement iPalace, as in Figure 1. In contrast, the challenges in outsourcing and diffusion activities, as in Figure 3, relate to interorganizational integration. NPM thus had to ensure video production and cloud computing could be outsourced appropriately.

Theoretically, RIIS diffusion is an organizational process involving participants from different industries and sectors across the value network. Although the aim of the service-science discipline is facilitating and improving interaction and collaboration of multiple entities to achieve mutual benefits, service science studies are generally summarized as “too much, too little, or too soon.”¹¹

Reflecting the museum visitor's perspective on iPalace, NPM seeks to gain a comprehensive understanding of the role, responsibilities, and involvement of every stakeholder associated with the iPalace value network. Moreover, iPalace delivers a viewing experience and post-viewing experience that must complement one another to ensure success.

With regard to the viewing process, the museum's online visitors desire an emotionally positive experience facilitated by a personal computer or portable computing device. As outlined in NPM's strategic vision for iPalace, fresh, attractive video content, well-defined NPM experiences, and smooth video streaming are necessary for ensuring online visitors have a positive experience online. NPM recognizes the content maker is responsible for producing fresh, attractive video content, the channel manager curates the video content and defines the NPM experience, and the connectivity maker ensures smooth video streaming.

With regard to the post-viewing experience, iPalace service quality directly correlates with viewer satisfaction, as with any commercial media experience in the real world. Service provision and the fulfillment of museum visitor needs are critical determinants of a viewer's use and enjoyment of iPalace. As a result, all parties in the iPalace value network, including NPM, content makers, connectivity makers, technology makers, and sponsors, must ensure effective viewer relationship management (VRM) and viewer fulfill-



Operation of the iPalace initiative would be like a television program that broadcasts continuously until the viewer turns off the channel.



ment. Note the emotive force of traditional television media leads viewers from brand awareness to brand consideration. This is where the process ends, and no other means extends into the post-viewing period. In contrast, agents (such as NPM, sponsors, and advertisers) are able to harness the reach and emotional power of iPalace and motivate viewers to complete the post-viewing process. This ability makes iPalace attractive to sponsors and advertisers and in turn to other service partners and stakeholders. During the post-viewing period, connectivity makers, sponsors, and advertisers are likely to risk loss of visitors' attention or failure to attract new visitors due to substandard VRM or insufficient viewer fulfillment.

Although iPalace theoretically consists of two processes, online visitors participate in both, yet regard them as a single, seamless experience. Online visitors expect a high-quality, consistently reliable service that safeguards user privacy. To ensure the success of iPalace technology, all participants in the iPalace value network must fulfill this expectation. However, service quality and reliability, as well as privacy, are particularly important for the channel manager and connectivity maker, who are likely to be deemed iPalace “providers” by museum visitors. The brands of the channel manager and connectivity maker may also be subject to negative evaluation if mistakes are made in service delivery; that is, online visitors contact the channel manager or the connectivity maker if iPalace delivers a disappointing experience. Museum visitors usually view brands that fail to fulfill VRM and viewer-fulfillment expectations negatively. As iPalace's VRM is Internet-based, it faces a particular threat of negative evaluation, as there are more than one billion Internet users (and thus potential museum visitors) worldwide.

All parties are thus responsible for determining the most appropriate way to carry iPalace through the value network, addressing several significant issues along the way:

Carrier-rights agreements. Carrier agreements, including channel sponsorship, are a key point of negotiation; in addition, connectivity makers must exercise caution when creating (or

outsourcing) VRM and viewer-fulfillment solutions;

Creative-rights agreements. Creative rights are a significant issue of contention among the various content makers. NPM's current distribution agreements do not include the right to develop an overlay on top of existing video content. Therefore, any future distribution agreements must include incentives for studios to allow channel managers to build iPalace enhancements; and

Sharing income from sponsorships. Facility expenditure, transformation expenditure, and market expenditure represent an ongoing burden for NPM and its service partners, requiring additional income from the service provided. NPM must address conflicts that involve other partners, including ownership of sponsorship income and how to share it fairly.

Conclusion

Garrison et al.⁶ said trust "between client organization and cloud provider is a strong predictor of a successful cloud deployment." It was therefore necessary for NPM to identify a suitable method of collaboration with other cloud actors, as well as with other business partners. NPM, a governmental entity, determined its own process of collaboration, which is largely regulated and conservative. NPM investigated the possibility of working with other cloud actors, including IBM, YouTube, Google, and domestic companies that have not yet established themselves as professional cloud actors. These alternatives presented strategic and managerial barriers, as well as different levels of technological readiness. Following several rounds of negotiation, the museum became more realistic about the future of iPalace and deployment of relevant complex technologies; for instance, due to the complicated political issues involved in collaborating with foreign actors, NPM identified a domestic value network as its first choice for helping develop museum artifacts and exhibitions. As Taiwan's cloud actors are not yet fully professional, NPM had to deal with a lack of technological readiness. In addition, NPM encountered conceptual differences relating to expenses, profit making, and trust when dealing with

certain domestic cloud actors; for instance, it did not wish to pursue commercialization due to the museum's not-for-profit status and operation, whereas most potential cloud actors aim to profit substantially from iPalace. NPM's current business model does not involve generating and sharing more than reasonable revenue. NPM has thus chosen to work with the National Center for High-performance Computing in Taiwan, as the Center's main objective is national technological advancement rather than pure profit making.

This study has several potential implications for managers of any traditional organization with less-advanced ICT expertise. To deploy and launch an RIIS, managers must establish and lead a radically innovative ICT-enabled service system. They must have in-depth understanding of all business partners and stakeholders involved in the embedded value network. They must negotiate carrier-rights agreements and creative-rights agreements. And they must also develop an effective business model (such as income sharing) to ensure success. Moreover, as in the case of iPalace, trust must also be nurtured and sustained.

Traditional organizations could also face other challenges, including conflicting laws and regulations that discourage development and implementation of an RIIS strategy; ineffective partners unable to provide expected service; inferior ICT infrastructure that makes new ICT-enabled service unattractive, thereby invalidating the whole project; and inadequate customer (or social) acceptance.

Market-related expenditure is also required to support social objectives (such as government support and social acceptance), interorganizational goals (such as satisfying key stakeholders, identifying trustworthy partners, maintaining good institutional governance, and understanding competitors' strategies), and managerial objectives (such as knowing the market and responding proactively).

The NPM experience provides other traditional organizations with lessons on how to deploy sophisticated technology; for instance, when an organization comprehensively implements an RIIS strategy to gain competitive advan-

tage, unresolved challenges like those described here could inhibit the RIIS and impede (and delay) the organization's evolution. On the other hand, the challenges associated with RIIS entail a valuable business opportunity for organizations able to implement an organization-centered, ICT-enabled, radically innovative value network.

Acknowledgments

We gratefully acknowledge financial support from the Ministry of Science and Technology, Taiwan, project numbers NSC 101-2420-H-004-005-MY3, NSC 102-2420-H-004-006-MY2, and MOST 103-2410-H-004-204. 

References

1. Bannon, L., Benford, S., Bowers, J., and Heath, C. Hybrid design creates innovative museum experiences. *Commun. ACM* 48, 3 (Mar. 2005), 62–65.
2. Bartak, A. The departing train: Online museum marketing in the age of engagement. In *Museum Marketing: Competing in the Global Marketplace*, R. Rentschler and A.-M. Hede, Eds. Butterworth-Heinemann, Oxford, U.K., 2007, 21–37.
3. Chang, W., Tsaih, R.H., Yen, D.C., and Han, T.S. *The ICT Predicament of New ICT-enabled Service*. Unpublished working paper, 2014; <http://arxiv.org/abs/1506.02128>
4. Chang, Y. *The Constitution and Understanding of Marketing Functions in the Museum Sector*. Unpublished Ph.D. thesis, King's College London, London, U.K., 2011; http://library.kcl.ac.uk/80/F/?func=direct&doc_number=001355984&local_base=KINGS.
5. Fitzsimmons, J.A. and Fitzsimmons, M.J. *Service Management: Operations, Strategy, Information Technology*. McGraw-Hill/Irwin, New York, 2008.
6. Garrison, G., Kim, S., and Wakefield, R.L. Success factors for deploying cloud computing. *Commun. ACM* 55, 9 (Sept. 2012), 62–68.
7. Hsi, S. and Falt, H. RFID enhances visitors' museum experience at the Exploratorium. *Commun. ACM* 48, 9 (Sept. 2005), 60–65.
8. Huang, S.Y., Chao, Y.T., and Tsaih, R.H. ICT-enabled service design suitable for museum: The case of the iPalace channel of the National Palace Museum in Taipei. *Journal of Library and Information Science* 39, 1 (Apr. 2013), 84–97.
9. International Council of Museums. *ICOM Definition of a Museum*, 2007; <http://icom.museum/definition.html>
10. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., and Leaf, D. *NIST Cloud Computing Reference Architecture*. National Institute of Standards and Technology Special Publication 500-292, Gaithersburg, MD, 2011; http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505
11. Spohrer, J. and Maglio, P.P. Service science: Toward a smarter planet. Chapter 1 in *Introduction to Service Engineering*, G. Salvendy and W. Karwowski, Eds. John Wiley & Sons, Inc., New York, 2010, 3–30.
12. vom Lehn, D. Generating aesthetic experiences from ordinary activity: New technology and the museum experience. Chapter 8 in *Marketing the Arts: A Fresh Approach*, D. O'Reilly and F. Kerrigan, Eds. Routledge, London, U.K., 2010, 104–120.

Rua-Huan Tsaih (tsaih@mis.nccu.edu.tw) is the vice dean of the Office of Research and Development of and a professor of MIS in the College of Commerce at the National Chengchi University in Taipei, Taiwan.

David C. Yen (david.yen@oneonta.edu) is the dean of and a professor in the School of Economics and Business at the State University of New York at Oneonta, Oneonta, NY.

Yu-Chien Chang (y.chang@nccu.edu.tw) is an assistant professor in the College of Commerce at the National Chengchi University in Taipei, Taiwan.

© 2015 ACM 00010782/15/08 \$15.00

Exploring three interdisciplinary areas and the extent to which they overlap. Are they all part of the same larger domain?

BY THANASSIS TIROPANIS, WENDY HALL, JON CROWCROFT,
NOSHIR CONTRACTOR, AND LEANDROS TASSIULAS

Network Science, Web Science, and Internet Science

THE OBSERVATION OF patterns that characterize networks, from biological to technological and social, and the impact of the Web and the Internet on society and business have motivated interdisciplinary research to advance our understanding of these systems. Their study has been the subject of Network Science research for a number of years. However, more recently we have witnessed the emergence of two new interdisciplinary areas: Web Science and Internet Science.

Network Science can be traced to its mathematical origins dating back to Leonard Euler's seminal work on graph theory¹⁵ in the 18th century and to its social scientific origins two centuries later by the psychiatrist

Jacob Moreno's²⁵ efforts to develop "sociometry." Soon thereafter, the mathematical framework offered by graph theory was also picked up by psychologists,² anthropologists,²³ and other social scientists to create an interdisciplinary called Social Networks. The interdisciplinary of Social Networks expanded even further toward the end of the 20th century with an explosion of interest in exploring networks in biological, physical, and technological systems. The term Network Science emerged as an interdisciplinary area that draws on disciplines such as physics, mathematics, computer science, biology, economics, and sociology to encompass networks that were not necessarily social.^{1,26,35} The study of networks involves developing explanatory models to understand the emergence of networks, building predictive models to anticipate the evolution of networks, and constructing prescriptive models to optimize the outcomes of networks. One of the main tenets of Network Science is to identify common underpinning principles and laws that apply across very different networks and explore why in some cases those patterns vary. The Internet and the Web, given their spectacular growth and impact, are networks that have captured the imagination of many network scientists.¹³ In addition, the emergence of

» key insights

- **Web Science and Internet Science aim to understand the evolution of the Web and the Internet respectively and to inform debates about their future. These goals lead to different priorities in their research agendas even though their communities overlap.**
- **Network Science aims to understand the evolution of networks regardless of where they emerge including: the Internet as a network transforming and forwarding information among people and things, and the Web as a network of creation and collaboration.**
- **Given their intellectual complementarities, we propose sharing and harmonizing the data research infrastructures being developed across these three interdisciplinary communities.**



online social networks and the potential to study online interactions on a massive, global scale hold the promise of further, potentially invaluable insights to network scientists on network evolution.²⁴

Web Science⁶ is an interdisciplinary area of much more recent vintage that studies the Web not only at the level of small technological innovations (micro level) but also as a phenomenon that affects societal and commercial activities globally (macro level); to a large extent, it can be considered the theory and practice of social machines on the Web. Social machines were conceptualized by Tim Berners-Lee in 1999 as artifacts where people do the creative work and machines intermediate.³ Semantic Web and linked data technologies can provide the means for knowledge representation and reasoning and enable further support for social machines.²⁰

Studying the Web and its impact requires an interdisciplinary approach that focuses not only on the technological level but also on the societal, political, and commercial levels. Establishing the relationship between these levels, understanding how they influence each other, investigating potential underpinning laws, and exploring ways to leverage this relationship in different domains of human activity is a large part of the Web Science research agenda. Web Science draws on disciplines that include the social sciences, such as anthropology, communication, economics, law, philosophy, political science, psychology, and sociology as well as computer sci-

ence and engineering. A major focus of the Web Science research agenda is to understand how the Web is evolving as a socio-technical phenomenon and how we can ensure it will continue to evolve and benefit society in the years to come.

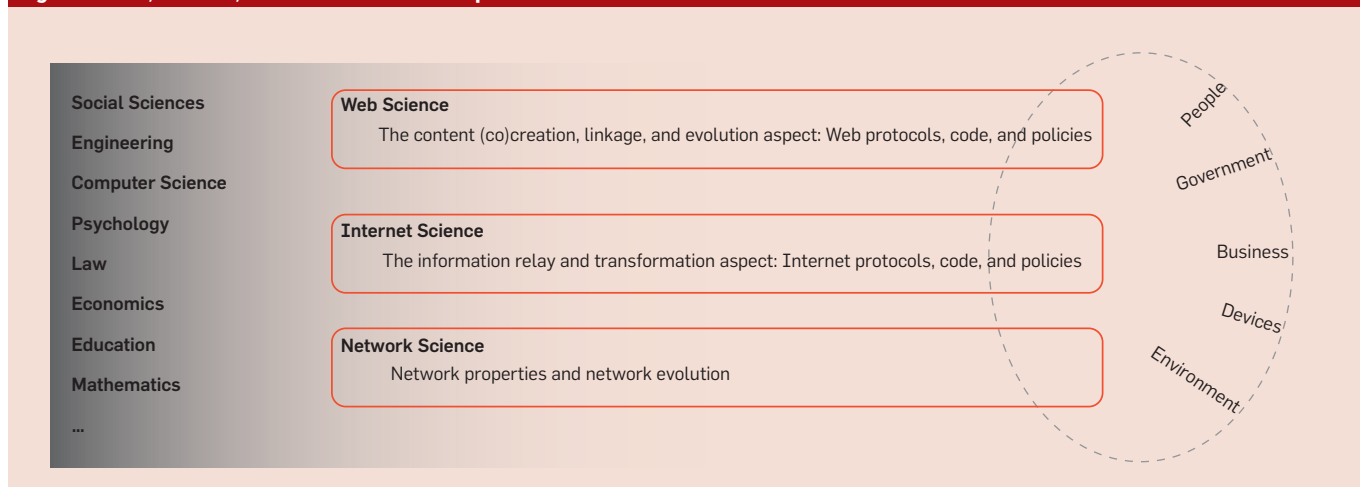
Internet Science. The Internet has provided the infrastructure on which much of human activity has become heavily dependent. After only a few decades of Internet development it is self-evident that if the Internet became unavailable, the consequences for society, commerce, the economy, defense, and government would be highly disruptive. The success of the Internet has often been attributed to its distributed governance model, the principle of network neutrality, and its openness.¹⁴ At the same time, concerns related to privacy, security, openness, and sustainability are raised and researched as they are often at the center of contestations on the Internet.¹¹ The Internet can be seen as an infrastructure, the social value of which must be safeguarded.¹⁸ It is the infrastructure that enabled the evolution of the Web along with P2P applications, more recently the cloud, and, in the near future, the Internet of Things. It has been argued the infrastructural layer of the Internet and that of the Web must be kept separately to foster innovation.⁴ A recent study⁷ identified a number of principled directions along which the Internet needs to evolve; those include availability, inclusiveness, scalability, sustainability, openness, security, privacy, and resilience. This motivates

the need for multidisciplinary research on Internet Science that seeks to understand the psychological, sociological, and economic implications of the Internet's evolution along these principled directions. Hence, Internet Science is an emerging interdisciplinary area that brings together scientists in network engineering, computation, complexity, security, trust, mathematics, physics, sociology, economics, political sciences, and law. This approach is very well exemplified by the early Internet topology study.¹⁶

Interdisciplinary relationships. All three areas draw on a number of disciplines for the study, respectively, of the nature and impact of the Web, of the Internet, and of networks in general on government, business, people, devices, and the environment. However, each of them examines how those actors co-create and evolve in distinct, unique ways as shown on Figure 1. For Web Science it is the aspect of linking those actors and the content with which they interact making associations between them and interpreting them. For Internet Science it is the aspect of communication among actors and resources as processes that can shape information relay and transformation. For Network Science, it is the aspect of how these entities, when considered to be part of a network, exhibit certain characteristics and might adhere to underpinning laws that can help understand their evolution.

However, to understand better the similarities and differences between these areas and to establish the po-

Figure 1. Web, Internet, and Network Science aspects.



tential for synergies, a framework for a more detailed comparison is needed.

A Comparison of Interdisciplinary Areas

It takes only a quick read through a short description of each of these interdisciplinary areas^{5,32,35} for one to realize that, to a very large extent, they all draw from very similar sets of disciplines. Venn diagrams that have been used to illustrate the involvement of different disciplines in each area are indicative of this overlap. For example, psychology and economics are considered relevant to Network Science,²⁹ Internet Science,⁷ and Web Science.²⁰ This can give rise to certain questions such as: “If there is so much overlap, aren’t these areas one and the same?” or “Would they all merge in the future?” Other questions include: “Which community is more relevant to my research?” or “What developments could we expect from each area in the future?” To explore those questions we propose a framework of examining those interdisciplinary areas, which includes looking at the way these communities have formed, and the different languages of discourse these communities have employed in their research.

Community formation. Although not all three interdisciplinary domains were established at the same time, one can argue that research in those areas dates back before their official starting date. At the same time, one can also argue there are differences in how communities around those domains emerged.

The formation of the Social Networks community can be traced back to a series of social network conferences that started in the 1970s¹⁷ with an important conference in Dartmouth in 1975 that brought together sociologists, anthropologists, social psychologists, and mathematicians from the U.S. and Europe. This was followed by Lin Freeman’s launch of *Social Networks* in 1978, and Barry Wellman founding the International Network for Social Network Analysis (INSNA) in 1976 and its annual Sunbelt Social Networks conference in 1981. Beginning in the 1990s, the social scientists were joined by a large and growing influx of scholars from the physical



All three areas draw on a number of disciplines for the study, respectively, of the nature and impact of the Web, of the Internet, and of networks in general on government, business, people, devices, and the environment.



and life sciences who began exploring networks in social systems. This effort was acknowledged and further catalyzed by the launch of the annual Network Science (NetSci) conference in 2006, a major infusion of funding in 2008 from the Army Research Laboratory for the development of an interdisciplinary Network Science Collaborative Technology Alliance (NS-CTA), and the launch of the *Network Science Journal* in 2013.^a Clearly there was already a community in place, which engaged in interdisciplinary work long before those initiatives; one can argue a hybrid bottom-up and top-down approach is the community formation model that was followed for Network Science.

For the Web Science community, it was around 2006 when it was realized that understanding the impact of the Web was essential to safeguard its development in the future. The Web Science Research Initiative (WSRI) was established in 2006 and later developed into the Web Science Trust (WST) as part of the top-down approach to the formation of the Web Science community. The WSRI raised a banner for those who were engaged in research on the Web as a socio-technical phenomenon, including the social network research community. A similar community formation model was followed for Internet Science, where the European Network of Excellence in Network Science (EINS)^{b,32} is one of the most significant activities to bring together the research community in this area. Areas such as privacy and network neutrality have been highlighted as priorities in the Internet Science agenda.

It can be argued the top-down model of community formation can accelerate research in emergent interdisciplinary areas but, in order to be successful, it requires a significant investment of resources from individuals, from research institutions, and from industry or government. Although the Web Science and the Internet Science communities were formed mostly in a top-down fashion, the sustainability of

a <http://journals.cambridge.org/action/displayJournal?jid=NWS>

b <http://www.internet-science.eu>

those communities was ensured by research funding from key research institutions, national research councils, the European Union, and significant effort by individuals.

Use of a lingua franca. Beyond community formation, there are differences in the language of discourse (the lingua franca) that is employed in each area. Network scientists initially shared graph theory as their lingua franca but have more recently employed models taken from physical processes (percolation, diffusion) and game theory¹³ to describe processes on graphs. They have also moved from descriptive network metrics to the development of novel inferential techniques to test hypotheses about the evolution of a network based on various self-organizing mechanisms.^{27,28} As a result, the use of graph theory is not necessarily the foundation for contemporary Network Science research. Further, there is use of complex systems analysis to deal with phase changes and discontinuities between different operating regimes; these are used to study why epidemics and pandemics spread globally. As a result, many Network Science publications are featured in journals such as *Nature*.

The Web Science community has not yet embraced a lingua franca per se but one can argue that an understanding of Web standards, technologies, and models (HTTP, XML, JavaScript, REST, models of communication, ontologies) and of frameworks of social theory are components of what could develop into a lingua franca. The W3C has been fostering a significant part of the discussion on Web protocols and their implications. A basic understanding of the evolution of the Web on both the micro and macro levels is the foundation for Web Science research.

Similar means of discourse are employed in the Internet Science community. For Internet Science, the components of the lingua franca include the set of Internet standards (RFCs) and associated commentary and implementation (or even C code) as in Stevens' books,^{30,31} as well as the existence of de facto standard implementations of systems in open source. They also include a basic

understanding of the principles of Internet protocols, infrastructure (routers, links, AS topology), social science (preferential attachment models), law, and policy.

Research methodologies. In Network Science, research methodologies involve network modeling and network analysis^{9,10} on networks that include, but are by no means restricted to, the Web and the Internet. In Internet Science, methodologies that employ measurements of engagement of Internet users with online resources and the Internet of Things are prevalent. In Web Science, mixed research methods that combine interpretative and positivist approaches are employed widely to understand the evolution of the Web based on online social network datasets, clickstream behavior, and the use of the Web data.

Beyond methodologies, the Web Science community is working on providing the Web Science Observatory,^{33,34} a global-distributed resource with datasets and analytic tools related to Web Science. Similarly, the EINS project is working on providing an evidence base for Internet Science research. And the Network Science community has a long tradition of making canonical network datasets available for use by the community along with network analysis software such as UCINET⁸ and large-scale repositories of network data such as SNAP.²²

Clearly there is an overlap in the research methodologies of these three areas:

- ▶ They draw on data gathered from social networks, infrastructures, sensors and the Internet of Things;
- ▶ They involve measurement, modeling, simulation, visualization, hypothesis testing, interpretation and exploratory research; and
- ▶ They use analytical techniques to quantify properties of a network (abstract, virtual, or real) as well as more qualitative techniques.

So far, there has been significant emphasis on the social sciences in Web Science, on both social science theories and methodologies in Network Science, and on protocols and computer science in Internet Science. However, these foci will change

in the future and the research methods or data will continue to mingle between these three areas. For example, data on the Internet of Things might not remain exclusive to Internet Science since that data could be combined with data on human behavior on the Web from the Web Science perspective or to explore emergence and outcomes of the networks they enable from the Network Science point of view. Similarly, data on the behavior of users on the Web will be used to explore the use of bandwidth in the underlying Internet infrastructure. The different types of measurement point to the fact that often, part of the research, especially in the top-down-formed areas of Internet Science and Web Science, is associated to specific goals.

Given this shared pool of methods and data resources, each area employs mixed methods to leverage this pool in different ways according to their research agendas as illustrated in Figure 2. Those agendas are informed by different research goals.

Research goals. "Web Science is focused on how we could do things better, while Network Science is more focused on how things work;"³⁶ the "doing things better" refers to leveraging the potential of the Web and ensuring its continuing sustainability. Similar claims are made on behalf of Internet Science and Network Science. Although the use of the term 'science' relates to the systematic organization of knowledge and is not directly linked to goals, we argue that goals do play a role in the formation of these interdisciplinary areas and in shaping their research agendas, scientific contribution, and impact. In Web Science, the study of the Web itself is crucial,²¹ as is safeguarding the Web and its evolution.¹⁹ In Internet Science, the evolution and sustainability of the Internet and its services are central objectives; it is understood that tussles will always be the case on the Internet and that accommodating them is necessary in order to ensure its evolution.¹¹ It seems that in both Internet Science and Web Science applied research comes first but it should be informed by the development of a basic research program. In addition, neither Web Science nor Internet Science is tech-

nology neutral; each one relies on specific protocols and standards. Further, one can argue that even the code that implements those standards embeds policy on which each respective community has reached some consensus. On the other hand, Network Science is technology agnostic and it overlaps only in part with Internet Science and Web Science, since it explores emergent structural patterns and flows on network structures be they social, biological, the Web, or the Internet. Finally, Web Science and Internet Science are both also engineering disciplines; they are about building better, stronger, more robust, efficient, and resilient systems. Network Science has been predominantly focused on understanding and describing emergent processes, although access to large datasets has increased interest in both predictive analytics to anticipate network changes and prescriptive analytics to optimize networks to accomplish certain desired goals. In essence, Network Science is aspiring to take insights from basic research to engineer better networks.¹²

Comparisons. Despite the differences between these areas in terms of the community formation models, lingua francas, and goals, many of the research methods they employ are common. This points to potential synergies on topics in which these areas overlap and the potential for mobilization within those communities on topics in which there is little or no overlap. Figure 3 shows such topics from each of these areas:

1. *Web Science:* The area of Web-based social media is one example of primarily Web Science research. Network aspects are not the exclusive part of this since social media research focuses on associations and interaction among people and social media resources.

2. *Internet Science:* Research on how the Internet of Things affects information collection and transformation is primarily Internet Science research that cannot rely exclusively on network research either.

3. *Network Science:* Transport networks provide an example of network science research that does not necessarily relate to Internet or Web science.

4. *Web Science and Internet Science:*

Network neutrality is an example that requires understanding of both Web and Internet technology and it does not necessarily draw primarily on network science techniques.

5. *Internet Science and Network Science:* Content Delivery Networks can

require network techniques for distribution prediction and optimization and, at the same time, understanding of how Internet protocols and people relate to shaping that demand.

6. *Network Science and Web Science:* Diffusion on social media such as

Figure 2. Network, Internet, and Web Science methodologies.

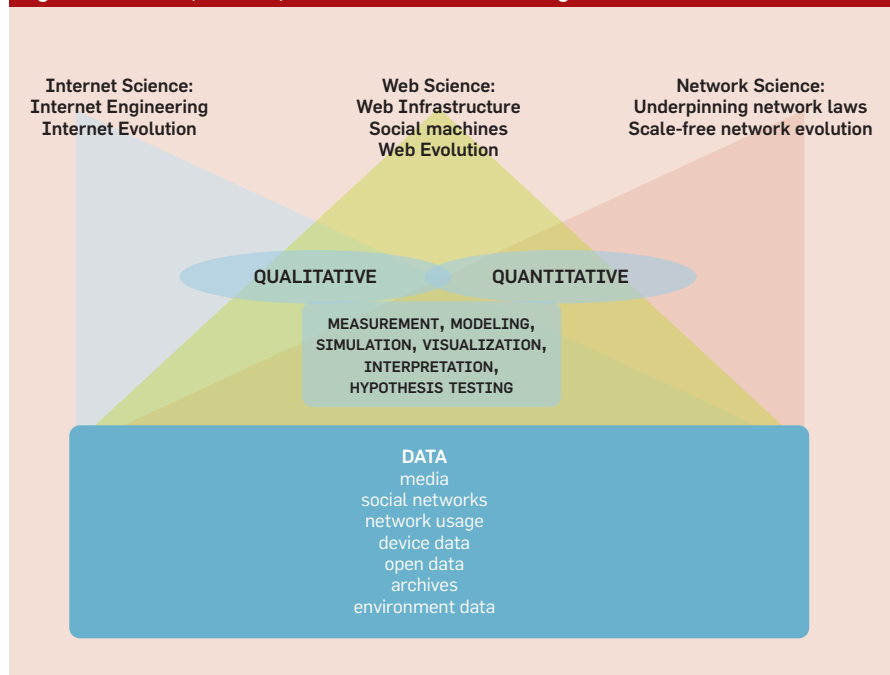
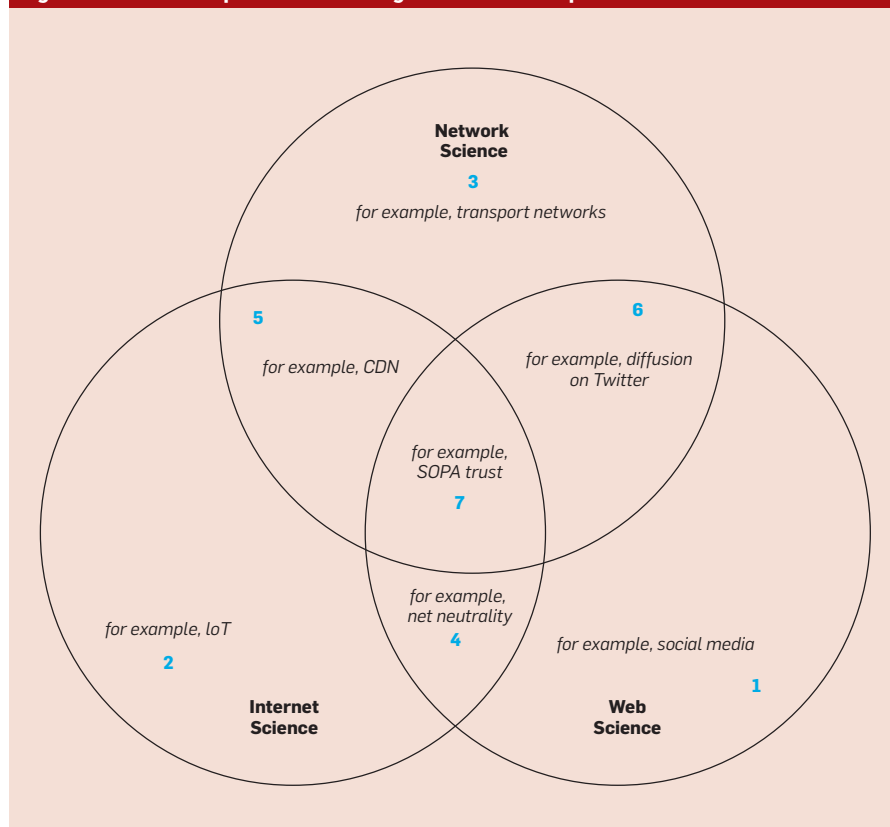


Figure 3. Research topics differentiating areas and overlaps.



Twitter is an example that relies on Web Science socio-technical research methods and, at the same time, on network analytic methods.

7. *Web Science, Internet Science, and Network Science*: Research on trust on-line or on SOPA (Stop Online Piracy Act) and its side effects draws on all networks and on techniques that are aware of Web and Internet protocols and code.

As the Web and the Internet continue to evolve it could be that some of these topics will shift.

Conclusion

We provide a comparison among Network, Web, and Internet Science. We also propose a framework for comparing interdisciplinary areas based on their community formation, lingua francas, research methods and resources, and research goals. We can gain additional insights of the relationship among these areas by conducting co-author and co-citation analysis of publications within these areas and explore the extent to which these are distinct or merging interdisciplinary intellectual communities. Such an analysis would be even more meaningful as the related conferences and journals mature and as the similarities and differences among these areas potentially crystallize.

Both Internet Science and Web Science are technology-aware and their respective lingua francas include knowledge of the protocols and systems supporting the Internet and the Web, while Network Science is technology-agnostic. There are arguments in keeping the two layers of the Internet and the Web separate to foster innovation;⁴ consequently, Internet Science and Web Science remain two distinct interdisciplinary areas given they have different goals, those of safeguarding the Internet and the Web, respectively. Network Science explores phenomena that include, but are not limited to, the Web or the Internet.

However, given the shared pool of mixed methods and datasets among these three interdisciplinary areas, there are compelling benefits for collaboration to harmonize and share resources; this should be a high priority for researchers and funding agencies.

Acknowledgments. The preparation of this manuscript was supported by funding from the U.S. Army Research Laboratory (9500010212/0013//W911NF-09-2-0053), the National Science Foundation (CNS-1010904) and the European Union FP7 Network of Excellence in Internet Science-EINS (grant agreement No 288021). The views, opinions, and/or findings contained here are those of the authors, and should not be construed as an official Department of the Army, NSF or European Commission position, policy, or decision, unless so designated by other documents. **C**

References

- Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.
- Bavelas, A. Communication patterns in task-oriented groups. *J. Acoustical Society of America* 22, 6 (1950), 725–730.
- Berners-Lee, T. *Weaving the Web*. Texere Publishing, 1999.
- Berners-Lee, T. Long live the Web. *Scientific American*, (2010).
- Berners-Lee, T., Hall, W., Hendler, J.A., O'Hara, K. and Shadbolt, N. A framework for Web science. *Foundations and Trends in Web Science* 1, 1 (2006b), 1–130.
- Berners-Lee, T., Hall, W., Hendler, J., Shadbolt, N. and Weitzner, D. Computer science enhanced: Creating a science of the Web. *Science* 313, 5788 (2006a), 769.
- Blackman, C., Brown, I., Cave, J., Forge, S., Guevara, K., Srivastava, L. and Popper, M.T.W.R. *Towards a Future Internet*, (2010). European Commission DG INFO Project SMART 2008/0049.
- Borgatti, S., Everett, M.G. and Freeman, L.C. *Computer Software: UCINET 6. Analytic Technologies*, 2006.
- Börner, K., Sanyal, S. and Vespignani, A. Network science. B. Cronin, Ed. *Annual Review of Information Science & Technology*, 41 (2007), 537–607.
- Carrington, P. J., Scott, J., and Wasserman, S., eds. *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005.
- Clark, D.D., Wroclawski, J., Sollins, K.R., and Braden, R. *Tussle in cyberspace: Defining tomorrow's Internet*. Aug. 2002. ACM.
- Contractor, N.S. and DeChurch, L.A. (in press). Integrating social networks and human social motives to achieve social influence at scale. In *Proceedings of the National Academy of Sciences*.
- Easley, D. and Kleinberg, J. *Networks Crowds and Markets*. Cambridge University Press, 2010.
- Economides, N. Net neutrality, non-discrimination and digital distribution of content through the Internet. *ISJLP* 4 (2008), 209.
- Euler, L. Konigsberg Bridge problem *Commentarii academiae scientiarum Petropolitanae* 8 (1741), 128–140.
- Faloutsos, F., Faloutsos, P. and Faloutsos, C. On power-law relationships of the Internet topology. *ACM SIGCOMM CCR*. Rev. 29, 4 (1999).
- Freeman, L.C. *The Development Of Social Network Analysis*. Booksurge LLC, 2004.
- Frischmann, B.M. *Infrastructure*. OUP USA, 2012.
- Hall, W. and Tiropanis, T. Web evolution and Web Science. *Computer Networks* 56, 18 (2012), 3859–3865.
- Hendler, J. and Berners-Lee, T. From the semantic web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence* (2009), 1–10.
- Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T. and Weitzner, D. Web Science: an interdisciplinary approach to understanding the Web. *Commun. ACM* 51, 7 (July 2008).
- Leskovec, J. Stanford large network dataset collection, 2011; <http://snap.stanford.edu/data/index.html>
- Mitchell, J.C. *The Kalela Dance: Aspects of Social Relationships among Urban Africans*. Manchester University Press, 1956.

- Monge, P.R. and Contractor, N.S. *Theories of Communication Networks*. Oxford University Press, 2003.
- Moreno, J.L. *Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Socio-Drama* (2nd ed.). Beacon House, Oxford, England, 1953.
- Newman, M.E.J. *Networks: An Introduction*. Oxford University Press, 2010.
- Robins, G., Snijders, T., Wang, P., Handcock, M., and Pattison, P. Recent developments in exponential random graph (p^*) models for social networks. *Social Networks* 29, 2 (2007), 192–215; doi:10.1016/j.socnet.2006.08.003
- Snijders, T.A.B., Van de Bunt, G.G., and Steglich, C.E.G. Introduction to stochastic actor-based models for network dynamics. *Social Networks* 32, 1 (2010), 44–60; doi:10.1016/j.socnet.2009.02.004
- Steen, M.V. Computer science, informatics, and the networked world. *Internet Computing, IEEE* 15, 3 (2011), 4–6.
- Stevens, W.R. *TCP/IP Illustrated, Vol. 1: The Protocols*, (1993).
- Stevens, W.R. and Wright, G.R. *TCP/IP Illustrated, Vol. 2: The Implementation*, (1995).
- The EINS Consortium. *EINS Factsheet*. The EINS Network of Excellence, EU-FP7 (2012); <http://www.internet-science.eu/publication/231>.
- Tiropanis, T., Hall, W., Hendler, J. de Larrinaga, Christian. The Web Observatory: A middle layer for broad data. *Dx.Doi.org* 2, 3 (2014), 129–133; doi:10.1089/big.2014.0035
- Tiropanis, T., Hall, W., Shadbolt, N., de Roure, D., Contractor, N. and Hendler, J. The Web Science Observatory. *Intelligent Systems, IEEE* 28, 2 (2013), 100–104.
- Watts, D. The 'new' science of networks. *Annual Review of Sociology*, (2004).
- Wright, A. Web science meets network science. *Commun. ACM* 54, 5 (May 2011).

Thanassis Tiropanis (t.tiropanis@southampton.ac.uk) is an associate professor with the Web and Internet Science Group, Electronics and Computer Science, University of Southampton, U.K.

Wendy Hall (wh@ecs.soton.ac.uk) is a professor of Computer Science at the University of Southampton, U.K. She is a former president of ACM.

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

Noshir Contractor (nosh@northwestern.edu) is the Jane S. & William J. White Professor of Behavioral Sciences in the McCormick School of Engineering & Applied Science, the School of Communication, and the Kellogg School of Management at Northwestern University, Chicago, IL.

Leandros Tassioulas (leandros.tassioulas@yale.edu) is the John C. Malone Professor of Electrical Engineering at Yale University, New Haven, CT.

research highlights

P. 84

Technical Perspective Corralling Crowd Power

By Aniket (Niki) Kittur

P. 85

Soylent: A Word Processor with a Crowd Inside

By Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann,
Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich

Technical Perspective

Corralling Crowd Power

By Aniket (Niki) Kittur

EARLY PIONEERS in computing, such as Herb Simon and Allen Newell, realized that human cognition could be framed in terms of information processing. Today, research like that described in the following paper is demonstrating the possibilities of seamlessly connecting human and machine information processors to accomplish creative tasks in ways previously unimaginable. This research is made possible by the rise of online crowdsourcing, in which millions of workers worldwide can be recruited for nearly any imaginable task. For some kinds of work and some fields of computer science these conditions have led to a renaissance in which large amounts of information are parallelized and labeled by human workers, generating unprecedented training sets for domains ranging from natural language processing to computer vision.

However, complex and creative tasks such as writing or design are not so straightforward to decompose and parallelize. Imagine, for example, a crowd of 100 workers let loose on your next paper with instructions to improve anything they find. You can quickly envision the challenges with coordinating crowds, including avoiding duplication of effort, dealing with conflicting viewpoints, and creating a system robust to any individual's lack of global context or expertise. Thus, the parallelized independent tasks typical of crowdsourcing today seem a poor match for the rich interactivity required for writing and editing tasks. These coordination and interactivity issues have been critical barriers to harnessing the power of crowds for complex and creative real-world tasks.

The authors introduce and realize an exciting vision of using crowd workers to power an interactive system—here, a word processor—in accomplishing complex cognitive tasks such as intelligently shortening text or acting as flexible “human macro.” This vision goes beyond previous “Wizard of Oz”-style approaches (in which humans are used to prototype functionality that is diffi-

cult to program) to permanently wiring human cognition into interactive systems. Such “crowd-powered systems” could enable the creation of entirely new forms of computational support not yet possible, and to build up training data that could help develop AI.

A central challenge in realizing this vision is coordinating crowds to accomplish interdependent tasks that cannot be easily decomposed; for example, a paragraph in which one sentence needs to flow into the next. The authors introduce a crowd programming pattern called *Find-Fix-Verify*, which breaks down tasks such that some workers identify areas that need transformation, others transform the most commonly identified areas, and others select the best transformations. Although no single individual need work on (or even read) the entire article, effort is focused into key areas while maintaining context within those areas. The authors show evidence that using this pattern crowds could collectively accomplish tasks with high-quality output—including shortening text, proofreading, and following open-ended instructions—despite relatively high individual error rates.

One might ask how such an approach scales in terms of time or complexity. In terms of time, crowd marketplaces can suffer from a latency problem in waiting for tasks to be accepted by workers, and indeed this accounted for the bulk of time in each condition (~20 minutes). In terms of complexity, the authors acknowledge an important limitation in the degree of interdependence supported; for example, changes requiring modification of large areas or related but separate areas can lead to quality issues.

However, the field (including the authors) has since made tremendous progress in scaling up the speed, quality, and complexity of crowd work. The time needed to recruit a crowd worker has dropped from minutes to seconds following the development of methods such as paying workers to be on

retainer, enabling time-sensitive applications such as helping blind users navigate their surroundings. The quality of crowd work has increased by orders of magnitude due to research ranging from improved task design (for example, using Bayesian Truth Serum where workers predict others' answers), to leveraging workers' behavioral traces (for example, looking at the way workers do their work instead of their output), to inferring worker quality across tasks and reweighting their influence accordingly.

Perhaps the most important question for the future of crowd work is whether it is capable of scaling up to the highly complex and creative tasks embodying the pinnacle of human cognition, such as science, art, and innovation. As the authors, myself, and others have argued (for example, in *The Future of Crowd Work*), doing so may be critical to enabling crowd workers to engage in the kinds of fulfilling, impactful work we would desire for our own children. Realizing this future will require highly interdisciplinary research into fundamental challenges ranging from incentive design to reputation systems to managing interdependent workflows. Such research will be complicated by but ultimately more impactful for grappling with the shifting landscape and ethical issues surrounding global trends towards decentralized work. Promisingly, there have been a number of recent examples of research using crowds to accomplish complex creative work including journalism, film animation, design critique, and even inventing new products. However, the best (or the worst) may be yet to come: we stand now at an inflection point where, with a concerted effort, computing research could tip us toward a positive future of crowd-powered systems. 

Aniket (Niki) Kittur is an associate professor and the Cooper-Siegel chair in the Human-Computer Interaction Institute at Carnegie Mellon University, Pittsburgh, PA.

Copyright held by author.

Soylent: A Word Processor with a Crowd Inside

By Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich

Abstract

This paper introduces architectural and interaction patterns for integrating crowdsourced human contributions directly into user interfaces. We focus on writing and editing, complex endeavors that span many levels of conceptual and pragmatic activity. Authoring tools offer help with pragmatics, but for higher-level help, writers commonly turn to other people. We thus present Soylent, a word processing interface that enables writers to call on Mechanical Turk workers to shorten, proofread, and otherwise edit parts of their documents on demand. To improve worker quality, we introduce the Find-Fix-Verify crowd programming pattern, which splits tasks into a series of generation and review stages. Evaluation studies demonstrate the feasibility of crowdsourced editing and investigate questions of reliability, cost, wait time, and work time for edits.

1. INTRODUCTION

Word processing is a complex task that touches on many goals of human-computer interaction. It supports a deep cognitive activity—writing—and requires complicated manipulations. Writing is difficult: even experts routinely make style, grammar, and spelling mistakes. Then, when a writer makes high-level decisions like changing a passage from past to present tense or fleshing out citation sketches into a true references section, she is faced with executing daunting numbers of nontrivial tasks across the entire document. Finally, when the document is a half-page over length, interactive software provides little support to help us trim those last few paragraphs. Good user interfaces aid these tasks; good artificial intelligence helps as well, but it is clear that we have far to go.

In our everyday life, when we need help with complex cognition and manipulation tasks, we often turn to other people. Writing is no exception⁵: we commonly recruit friends and colleagues to help us shape and polish our writing. But we cannot always rely on them: colleagues do not want to proofread every sentence we write, cut a few lines from every paragraph in a 10-page paper, or help us format 30 ACM-style references.

Soylent is a word processing interface that utilizes crowd contributions to aid complex writing tasks ranging from error prevention and paragraph shortening to automation of tasks such as citation searches and tense changes. Using *Soylent* is like having an entire editorial staff available as you write. We hypothesize that crowd workers with a basic knowledge of written English can support both novice and expert writers. These workers perform tasks that the writer might not, such as scrupulously scanning for text to cut or updating a list of addresses to include a zip code. They can

also solve problems that artificial intelligence cannot yet, for example flagging writing errors that the word processor does not catch.

Soylent aids the writing process by integrating paid crowd workers from Amazon's Mechanical Turk platform into Microsoft Word. *Soylent is people*^a: its core algorithms involve calls to Mechanical Turk workers (Turkers). *Soylent* is comprised of three main components:

1. *Shortn*, a text shortening service that cuts selected text down to 85% of its original length on average without changing the meaning of the text or introducing writing errors.
2. *Crowdproof*, a human-powered spelling and grammar checker that finds problems Word misses, explains the error, and suggests fixes.
3. *The Human Macro*, an interface for offloading arbitrary word processing tasks such as formatting citations or finding appropriate figures.

The main contribution of *Soylent* is *the idea of embedding paid crowd workers in an interactive user interface to support complex cognition and manipulation tasks on demand*. This paper contributes the design of one such system, an implementation embedded in Microsoft Word, and a programming pattern that increases the reliability of paid crowd workers on complex tasks. It then expands these contributions with feasibility studies of the performance, cost, and time delay of our three main components and a discussion of the limitations of our approach with respect to privacy, delay, cost, and domain knowledge.

The fundamental technical contribution of this system is a crowd programming pattern called *Find-Fix-Verify*. Mechanical Turk costs money and it can be error-prone; to be worthwhile to the user, we must control costs and ensure correctness. *Find-Fix-Verify* splits complex crowd intelligence tasks into a series of generation and review stages that utilize independent agreement and voting to produce reliable results. Rather than ask a single crowd worker to read and edit an entire paragraph, for example, *Find-Fix-Verify* recruits one set of workers to find candidate areas for improvement, another set to suggest improvements to those candidates,

^a With apologies to Charlton Heston (1973): *Soylent is made out of people*.

A full version of this paper was published in *Proceedings of ACM UIST 2010*.

and a final set to filter incorrect candidates. This process prevents errant crowd workers from contributing too much or too little, or introducing errors into the document.

In the rest of this paper, we introduce Soylent and its main components: Shortn, Crowdproof, and The Human Macro. We detail the Find-Fix-Verify pattern that enables Soylent, then evaluate the feasibility of Find-Fix-Verify and our three components.

2. RELATED WORK

Soylent is related to work in two areas: crowdsourcing systems and artificial intelligence for word processing.

2.1. Crowdsourcing

Gathering data to train algorithms is a common use of crowdsourcing. For example, the ESP Game¹⁹ collects descriptions of objects in images for use in object recognition. Mechanical Turk is already used to collect labeled data for machine vision¹⁸ and natural language processing.¹⁷ Soylent tackles problems that are currently infeasible for AI algorithms, even with abundant data. However, Soylent's output may be used to train future AIs.

Soylent builds on work embedding on-demand workforces inside applications and services. For example, Amazon Remembers uses Mechanical Turk to find products that match a photo taken by the user on a phone, and PEST¹⁶ uses Mechanical Turk to vet advertisement recommendations. These systems consist of a single user operation and little or no interaction. Soylent extends this work to more creative, complex tasks where the user can make personalized requests and interact with the returned data by direct manipulation.

Soylent's usage of human computation means that its behavior depends in large part on qualities of crowdsourcing systems and Mechanical Turk in particular. Ross et al. found that Mechanical Turk had two major populations: well-educated, moderate-income Americans, and young, well-educated but less wealthy workers from India.¹⁵ Kittur and Chi⁸ considered how to run user studies on Mechanical Turk, proposing the use of quantitative verifiable questions as a verification mechanism. Find-Fix-Verify builds on this notion of requiring verification to control quality. Heer and Bostock⁶ explored Mechanical Turk as a testbed for graphical perception experiments, finding reliable results when they implemented basic measures like qualification tests. Little et al.¹¹ advocate the use of human computation algorithms on Mechanical Turk. Find-Fix-Verify may be viewed as a new design pattern for human computation algorithms. It is specifically intended to control lazy and overeager Turkers, identify which edits are tied to the same problem, and visualize them in an interface. Quinn and Bederson¹⁴ have authored a survey of human computation systems that expands on this brief review.

2.2. Artificial intelligence for word processing

Soylent is inspired by writers' reliance on friends and colleagues to help shape and polish their writing.⁵

Proofreading is emerging as a common task on Mechanical Turk. Standard Minds^b offers a proofreading service backed by Mechanical Turk that accepts plain text via a web form and returns edits 1 day later. By contrast, Soylent is embedded in a word processor, has much lower latency, and presents the edits in Microsoft Word's user interface. Our work also contributes the Find-Fix-Verify pattern to improve the quality of such proofreading services.

Automatic proofreading has a long history of research⁹ and has seen successful deployment in word processors. However, Microsoft Word's spell checker frequently suffers from false positives, particularly with proper nouns and unusual names. Its grammar checker suffers from the opposite problem: it misses blatant errors.^c Human checkers are currently more reliable, and can also offer suggestions on how to fix the errors they find, which is not always possible for Word—for example, consider the common (but mostly useless) Microsoft Word feedback, "Fragment; consider revising."

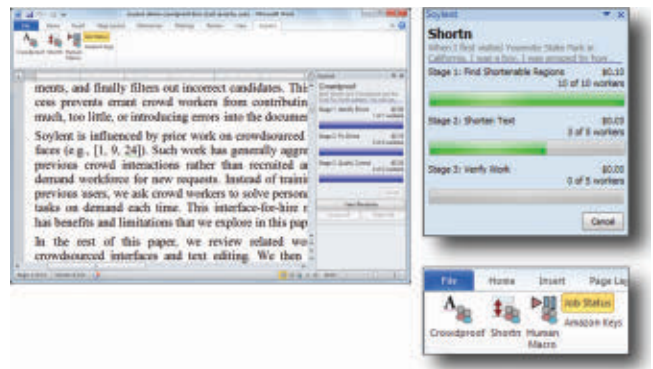
Soylent's text shortening component is related to document summarization, which has also received substantial research attention.¹² Microsoft Word has a summarization feature that uses sentence extraction, which identifies whole sentences to preserve in a passage and deletes the rest, producing substantial shortening but at a great cost in content. Shortn's approach, which can rewrite or cut parts of sentences, is an example of sentence compression, an area of active research that suffers from a lack of training data.³ Soylent's results produce training data to help push this research area forward.

The Human Macro is related to AI techniques for end-user programming. Several systems allow users to demonstrate repetitive editing tasks for automatic execution; examples include Eager, TELS, and Cima.⁴

3. SOYLENT

Soylent is a prototype crowdsourced word processing interface. It is currently built into Microsoft Word (Figure 1), a popular word processor and productivity application.

Figure 1. Soylent adds a set of crowd-powered commands to the word processor.



^b <http://standardminds.com/>.

^c <http://faculty.washington.edu/sandeep/check>.

It demonstrates that computing systems can reach out to crowds to: (1) create new kinds of interactive support for text editing, (2) extend artificial intelligence systems such as style checking, and (3) support natural language commands. These three goals are embedded in Soylent's three main features: text shortening, proofreading, and arbitrary macro tasks.

3.1. Shortn: Text shortening

Shortn aims to demonstrate that crowds can support new kinds of interactions and interactive systems that were very difficult to create before. Some authors struggle to remain within length limits on papers and spend the last hours of the writing process tweaking paragraphs to shave a few lines. This is painful work and a questionable use of the authors' time. Other writers write overly wordy prose and need help editing. Automatic summarization algorithms can identify relevant subsets of text to cut.¹² However, these techniques are less well-suited to small, local language tweaks like those in Shortn, and they cannot guarantee that the resulting text flows well.

Soylent's Shortn interface allows authors to condense sections of text. The user selects the area of text that is too long—for example, a paragraph or section—then presses the Shortn button in Word's Soylent command tab (Figure 1). In response, Soylent launches a series of Mechanical Turk tasks in the background and notifies the user when the text is ready. The user can then launch the Shortn dialog box (Figure 2). On the left is the original paragraph; on the right is the proposed revision. Shortn provides a single slider to allow the user to continuously adjust the length of the paragraph. As the user does so, Shortn computes the combination of crowd trimmings that most closely match the desired length and presents that text to the user on the right. From the user's point of view, as she moves the slider to make the paragraph shorter, sentences are slightly edited, combined

and cut completely to match the length requirement. Areas of text that have been edited or removed are highlighted in red in the visualization. These areas may differ from one slider position to the next.

Shortn typically can remove up to 15%–30% of a paragraph in a single pass, and up to 50% with multiple iterations. It preserves meaning when possible by encouraging crowd workers to focus on wordiness and separately verifying that the rewrite does not change the user's intended meaning. Removing whole arguments or sections is left to the user.

3.2. Crowdproof: Crowdsourced copyediting

Shortn demonstrates that crowds can power new kinds of interactions. We can also involve crowds to augment the artificial intelligence built into applications, for example proofreading. Crowdproof instantiates this idea.

Soylent provides a human-aided spelling, grammar, and style checking interface called Crowdproof (Figure 3). The process finds errors, explains the problem, and offers one to five alternative rewrites. Crowdproof is essentially a distributed proofreader or copyeditor.

To use Crowdproof, the user highlights a section of text and presses the proofreading button in the Soylent ribbon tab. The task is queued to the Soylent status pane and the user is free to keep working. Because Crowdproof costs money, it does not issue requests unless commanded.

When the crowd is finished, Soylent calls out the erroneous sections with a purple dashed underline. If the user clicks on the error, a drop-down menu explains the problem and offers a list of alternatives. By clicking on the desired alternative, the user replaces the incorrect text with an option of his or her choice. If the user hovers over the Error Descriptions menu item, the popout menu suggests additional second-opinions of why the error was called out.

Figure 2. Shortn allows users to adjust the length of a paragraph via a slider. Red text indicates locations where the crowd has provided a rewrite or cut. Tick marks on the slider represent possible lengths.

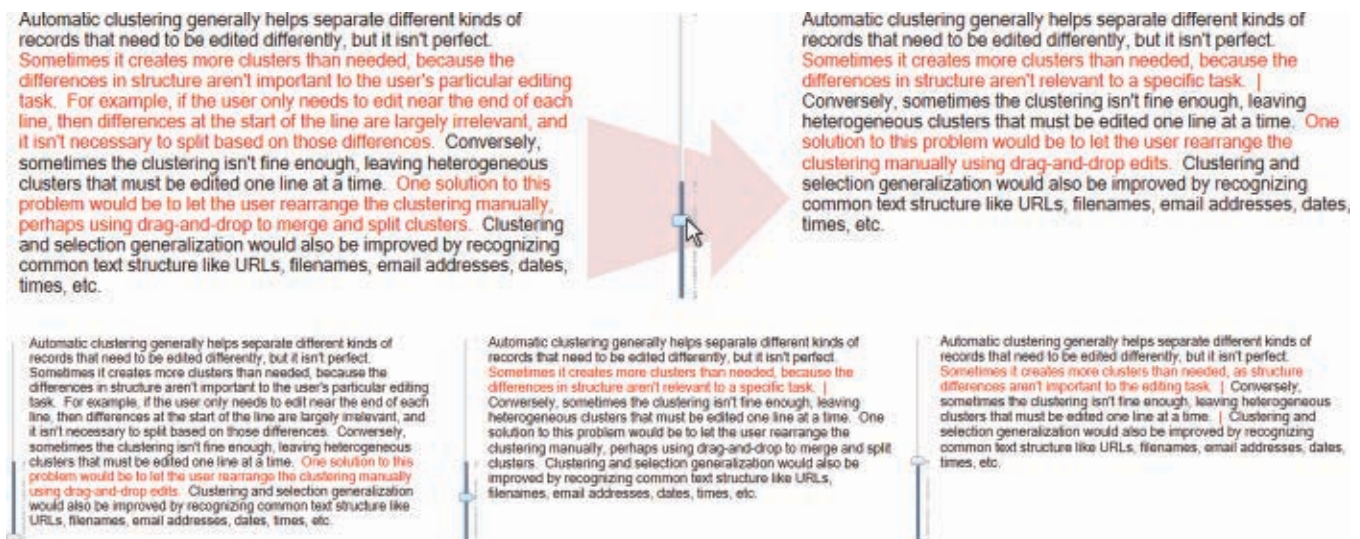
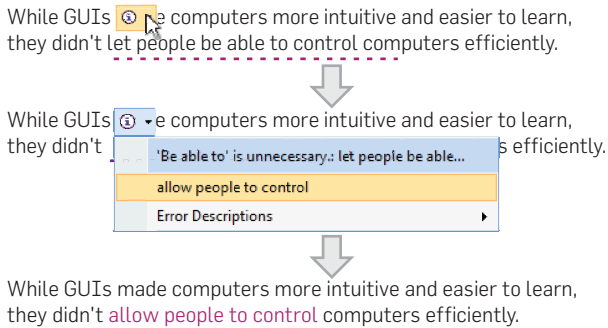


Figure 3. Crowdproof is a human-augmented proofreader. The drop-down explains the problem (blue title) and suggests fixes (gold selection).



3.3. The human macro: Natural language crowd scripting

Embedding crowd workers in an interface allows us to reconsider designs for short end-user programming tasks. Typically, users need to translate their intentions into algorithmic thinking explicitly via a scripting language or implicitly through learned activity.⁴ But tasks conveyed to humans can be written in a much more natural way. While natural language command interfaces continue to struggle with unconstrained input over a large search space, humans are good at understanding written instructions.

The Human Macro is Soylent's natural language command interface. Soylent users can use it to request arbitrary work quickly in human language. Launching the Human Macro opens a request form (Figure 4). The design challenge here is to ensure that the user creates tasks that are scoped correctly for a Mechanical Turk worker. We wish to prevent the user from spending money on a buggy command.

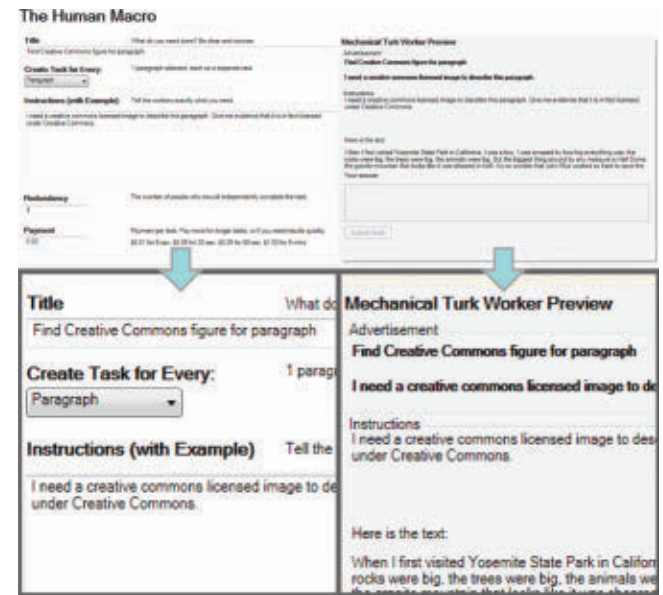
The form dialog is split in two mirrored pieces: a task entry form on the left, and a preview of what the crowd worker will see on the right. The preview contextualizes the user's request, reminding the user that they are writing something akin to a Help Wanted or Craigslist advertisement. The form suggests that the user provide an example input and output, which is an effective way to clarify the task requirements to workers. If the user selected text before opening the dialog, they have the option to split the task by each sentence or paragraph, so (for example) the task might be parallelized across all entries on a list. The user then chooses how many separate workers he would like to complete the task. The Human Macro helps debug the task by allowing a test run on one sentence or paragraph.

The user chooses whether the crowd workers' suggestions should replace the existing text or just annotate it. If the user chooses to replace, the Human Macro underlines the text in purple and enables drop-down substitution like the Crowdproof interface. If the user chooses to annotate, the feedback populates comment bubbles anchored on the selected text by utilizing Word's reviewing comments interface.

4. PROGRAMMING WITH CROWDS

This section characterizes the challenges of leveraging crowd labor for open-ended document editing tasks. We introduce

Figure 4. The Human Macro allows users to request arbitrary tasks over their document. Left: user's request pane. Right: crowd worker task preview, which updates as the user edits the request pane.



the Find-Fix-Verify pattern to improve output quality in the face of uncertain crowd worker quality. As we prepared Soylent and explored the Mechanical Turk platform, we performed and documented dozens of experiments.⁴ For this project alone, we have interacted with over 10,000 workers across over 2,500 different tasks. We draw on this experience in the sections to follow.

4.1. Challenges

We are primarily concerned with tasks where crowd workers directly edit a user's data in an open-ended manner. These tasks include shortening, proofreading, and user-requested changes such as address formatting. In our experiments, it is evident that many of the raw results that workers produce on such tasks are unsatisfactory. As a rule-of-thumb, roughly 30% of the results from open-ended tasks are poor. This "30% rule" is supported by the experimental section of this paper as well. Clearly, a 30% error rate is unacceptable to the end user. To address the problem, it is important to understand the nature of unsatisfactory responses.

High variance of effort. Crowd workers exhibit high variance in the amount of effort they invest in a task. We might characterize two useful personas at the ends of the effort spectrum, the *Lazy Turker* and the *Eager Beaver*. The *Lazy Turker* does as little work as necessary to get paid. For example, we asked workers to proofread the following error-filled paragraph from a high school essay site.⁵ Ground-truth errors are colored below, highlighting some of the low quality elements of the writing:

⁴ <http://groups.csail.mit.edu/uid/deneme/>.

⁵ <http://www.essay.org/school/english/ofmicelandmen.txt>.

The theme of loneliness features throughout many scenes in *Of Mice and Men* and is often the dominant theme of sections during this story. This theme occurs during many circumstances but is not present from start to finish. In my mind for a theme to be pervasive is must be present during every element of the story. There are many themes that are present most of the way through such as sacrifice, friendship and comradeship. But in my opinion there is only one theme that is present from beginning to end, this theme is pursuit of dreams.

However, a Lazy Turker inserted only a single character to correct a spelling mistake. The single change is highlighted below:

The theme of loneliness features throughout many scenes in *Of Mice and Men* and is often the dominant theme of sections during this story. This theme occurs during many circumstances but is not present from start to finish. In my mind for a theme to be pervasive is must be present during every element of the story. There are many themes that are present most of the way through such as sacrifice, friendship and comrade-ship. But in my opinion there is only one theme that is present from beginning to end, this theme is pursuit of dreams.

This worker fixed the spelling of the word *comradeship*, leaving many obvious errors in the text. In fact, it is not surprising that the worker chose to make this edit, since it was the only word in the paragraph that would have been underlined in their browser because it was misspelled. A first challenge is thus to discourage workers from exhibiting such behavior.

Equally problematic as Lazy Turkers are Eager Beavers. Eager Beavers go beyond the task requirements in order to be helpful, but create further work for the user in the process. For example, when asked to reword a phrase, one Eager Beaver provided a litany of options:

The theme of loneliness features throughout many scenes in *Of Mice and Men* and is often the principal, significant, primary, preeminent, prevailing, foremost, essential, crucial, vital, critical theme of sections during this story.

In their zeal, this worker rendered the resulting sentence ungrammatical. Eager Beavers may also leave extra comments in the document or reformat paragraphs. It would be problematic to funnel such work back to the user.

Both the Lazy Turker and the Eager Beaver are looking for a way to clearly signal to the requester that they have completed the work. Without clear guidelines, the Lazy Turker will choose the path that produces any signal and the Eager Beaver will produce too many signals.

Crowd workers introduce errors. Crowd workers attempting complex tasks can accidentally introduce substantial new errors. For example, when proofreading paragraphs about the novel *Of Mice and Men*, workers variously changed the title to just *Of Mice*, replaced existing grammar errors with new errors of their own, and changed the text to state that *Of Mice and Men* is a movie rather than a novel. Such errors are compounded if the output of one worker is used as input for other workers.

The result: Low-quality work. These issues compound into what we earlier termed the 30% rule: that roughly one-third of the suggestions we get from workers on Mechanical Turk are not high-enough quality to show an end user. We cannot simply ask workers to help shorten or proofread a paragraph: we need to guide and coordinate their activities.

These two personas are not particular to Mechanical Turk. Whether we are using intrinsic or extrinsic motivators—money, love, fame, or others—there is almost always an un-even distribution of participation. For example, in Wikipedia, there are many Eager Beaver editors who try hard to make edits, but they introduce errors along the way and often have their work reverted.

4.2. The Find-Fix-Verify pattern

Crowd-powered systems must control the efforts of both the Eager Beaver and Lazy Turker and limit the introduction of errors. Absent suitable control techniques for open-ended tasks, the rate of problematic edits is too high to be useful. We feel that the state of programming crowds is analogous to that of UI technology before the introduction of design patterns like Model-View-Controller, which codified best practices. In this section, we propose the Find-Fix-Verify pattern as one method of programming crowds to reliably complete open-ended tasks that directly edit the user's data.^f We describe the pattern and then explain its use in Solyent across tasks like proofreading and text shortening.

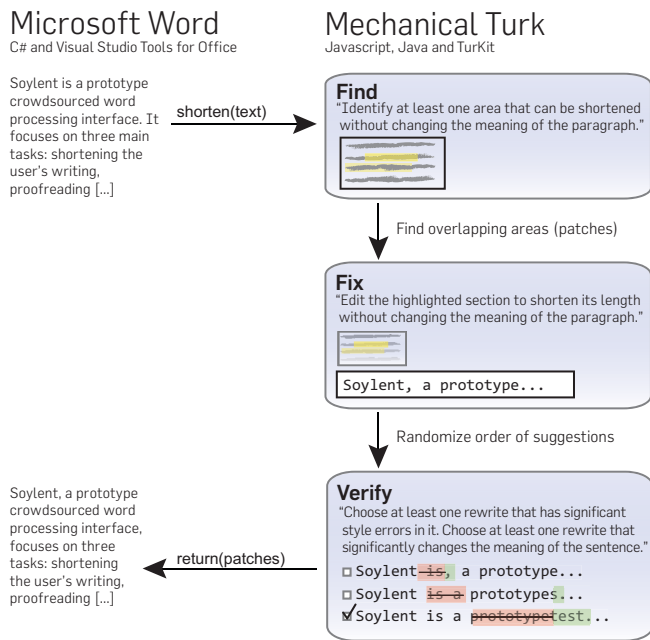
Find-Fix-Verify description. The Find-Fix-Verify pattern separates open-ended tasks into three stages where workers can make clear contributions. The workflow is visualized in Figure 5.

Both Shortn and Crowdproof use the Find-Fix-Verify pattern. We will use Shortn as an illustrative example in this section. To provide the user with near-continuous control of paragraph length, Shortn should produce many alternative rewrites without changing the meaning of the original text or introduce^g grammatical errors. We begin by splitting the input region into paragraphs.

^f Closed-ended tasks like voting can test against labeled examples for quality control.¹⁰ Open-ended tasks have many possible correct answers, so gold standard voting is less useful.

^g Word's grammar checker, eight authors and six reviewers on the original Solyent paper did not catch the error in this sentence. Crowdproof later did, and correctly suggested that "introduce" should be "introducing."

Figure 5. Find-Fix-Verify identifies patches in need of editing, recruits crowd workers to fix the patches, and votes to approve work.



The first stage, Find, asks several crowd workers to identify patches of the user's work that need more attention. For example, when shortening, the Find stage asks ten workers for at least one phrase or sentence that needs to be shortened. Any single worker may produce a noisy result (e.g., Lazy Turkers might prefer errors near the beginning of a paragraph). The Find stage aggregates independent opinions to find the most consistently cited problems: multiple independent agreement is typically a strong signal that a crowd is correct. Soylent keeps patches where at least 20% of the workers agree. These are then fed in parallel into the Fix stage.

The Fix stage recruits workers to revise each agreed-upon patch. Each task now consists of a constrained edit to an area of interest. Workers see the patch highlighted in the paragraph and are asked to fix the problem (e.g., shorten the text). The worker can see the entire paragraph but only edit the sentences containing the patch. A small number (3–5) of workers propose revisions. Even if 30% of work is bad, 3–5 submissions are sufficient to produce viable alternatives. In Shortn, workers also vote on whether the patch can be cut completely. If so, we introduce the empty string as a revision.

The Verify stage performs quality control on revisions. We randomize the order of the unique alternatives generated in the Fix stage and ask 3–5 new workers to vote on them (Figure 5). We either ask workers to vote on the best option (when the interface needs a default choice, like Crowdproof) or to flag poor suggestions (when the interface requires as many options as possible, like Shortn). To ensure that workers cannot vote for their own work, we ban all Fix workers from participating in the Verify stage for that paragraph. To aid comparison, the Mechanical Turk task annotates each rewrite using color and strikethroughs to

highlight its differences from the original. We use majority voting to remove problematic rewrites and to decide if the patch can be removed. At the end of the Verify stage, we have a set of candidate patches and a list of verified rewrites for each patch.

To keep the algorithm responsive, we use a 15-minute timeout at each stage. If a stage times out, we still wait for at least six workers in Find, three workers in Fix, and three workers in Verify.

Pattern discussion. Why should tasks be split into independent Find-Fix-Verify stages? Why not let crowd workers find an error and fix it, for increased efficiency and economy? Lazy Turkers will always choose the easiest error to fix, so combining Find and Fix will result in poor coverage. By splitting Find from Fix, we can direct Lazy Turkers to propose a fix to patches that they might otherwise ignore. Additionally, splitting Find and Fix enables us to merge work completed in parallel. Had each worker edited the entire paragraph, we would not know which edits were trying to fix the same problem. By splitting Find and Fix, we can map edits to patches and produce a much richer user interface—for example, the multiple options in Crowdproof's replacement dropdown.

The Verify stage reduces noise in the returned result. The high-level idea here is that we are placing the workers in productive tension with one another: one set of workers is proposing solutions, and another set is tasked with looking critically at those suggestions. Anecdotally, workers are better at vetting suggestions than they are at producing original work. Independent agreement among Verify workers can help certify an edit as good or bad. Verification trades off time lag with quality: a user who can tolerate more error but needs less time lag might opt not to verify work or use fewer verification workers.

Find-Fix-Verify has downsides. One challenge that the Find-Fix-Verify pattern shares with other Mechanical Turk algorithms is that it can stall when workers are slow to accept the task. Rather than wait for ten workers to complete the Find task before moving on to Fix, a timeout parameter can force our algorithm to advance if a minimum threshold of workers have completed the work. Find-Fix-Verify also makes it difficult for a particularly skilled worker to make large changes: decomposing the task makes it easier to complete for the average worker, but may be more frustrating for experts in the crowd.

5. IMPLEMENTATION

Soylent consists of a front-end application-level add-in to Microsoft Word and a back-end service to run Mechanical Turk tasks (Figure 5). The Microsoft Word plug-in is written using Microsoft Visual Studio Tools for Office (VSTO) and the Windows Presentation Foundation (WPF). Back-end scripts use the TurKit Mechanical Turk toolkit.¹¹

Shortn in particular must choose a set of rewrites when given a candidate slider length. When the user specifies a desired maximum length, Shortn searches for the longest combination of rewrites subject to the length constraint. A simple implementation would exhaustively list all combinations and then cache them, but this approach scales poorly with many patches. If runtime becomes an issue, we can view the search as a multiple-choice knapsack problem. In a

multiple-choice knapsack problem, the items to be placed into the knapsack come from multiple classes, and only one item from each class may be chosen. So, for Shortn, each item class is an area of text with one or more options: each class has one option if it was not selected as a patch, and more options if the crowd called out the text as a patch and wrote alternatives. The multiple-choice knapsack problem can be solved with a polynomial time dynamic programming algorithm.

6. EVALUATION

Our initial evaluation sought to establish evidence for SoyLent’s end-to-end feasibility, as well as to understand the properties of the Find-Fix-Verify design pattern.

6.1. Shortn evaluation

We evaluated Shortn quantitatively by running it on example texts. Our goal was to see how much Shortn could shorten text, as well as its associated cost and time characteristics. We collected five examples of texts that might be sent to Shortn, each between one and seven paragraphs long. We chose these inputs to span from preliminary drafts to finished essays and from easily understood to dense technical material (Table 1).

To simulate a real-world deployment, we ran the algorithms with a timeout enabled and set to 20 minutes for each stage. We required 6–10 workers to complete the Find tasks and 3–5 workers to complete the Fix and Verify tasks: if a Find task failed to recruit even six workers, it might wait

indefinitely. To match going rates on Mechanical Turk, we paid \$0.08 per Find, \$0.05 per Fix, and \$0.04 per Verify.

Each resulting paragraph had many possible variations depending on the number of shortened alternatives that passed the Verify stage—we chose the shortest possible version for analysis and compared its length to the original paragraph. We also measured *wait time*, the time between posting the task and the worker accepting the task, and *work time*, the time between acceptance and submission. In all tasks, it was possible for the algorithm to stall while waiting for workers, having a large effect on averages. Therefore, we report medians, which are more robust to outliers.

Results. Shortn produced revisions that were 78%–90% of the original document length. For reference, a reduction to 85% could slim an 11 $\frac{3}{4}$ page ACM paper draft down to 10 pages with no substantial cuts in the content. Table 1 summarizes and gives examples of Shortn’s behavior. Typically, Shortn focused on unnecessarily wordy phrases like “are going to have to” (Table 1, Blog). Crowd workers merged sentences when patches spanned sentence boundaries (Table 1, Classic UIST Paper), and occasionally cut whole phrases or sentences.

To investigate time characteristics, we separate the notion of wait time from work time. The vast majority of Shortn’s running time is currently spent waiting, because it can take minutes or hours for workers to find and accept the task. Here, our current wait time—summing the median Find, median Fix, and median Verify—was 18.5 minutes (1st Quartile Q_1 = 8.3 minutes, 3rd Quartile Q_3 = 41.6 minutes).

Table 1. Our evaluation run of Shortn produced revisions between 78%–90% of the original paragraph length on a single run.

Input text	Original length	Final length	Work stats	Time per paragraph (min)	Example Output
Blog	3 paragraphs, 12 sentences, 272 words	83% character length	\$4.57, 158 workers	46–57	Print publishers are in a tizzy over Apple’s new iPad because they hope to finally be able to charge for their digital editions. But in order to get people to pay for their magazine and newspaper apps, they are going to have to offer something different that readers cannot get at the newsstand or on the open Web.
Classic UIST Paper ⁷	7 paragraphs, 22 sentences, 478 words	87%	\$7.45, 264 workers	49–84	The metaDESK effort is part of the larger Tangible Bits project: The Tangible Bits vision paper, which introduced the metaDESK along with and two companion platforms, the transBOARD and ambient ROOM.
Draft UIST Paper	5 paragraphs, 23 sentences, 652 words	90%	\$7.47, 284 workers	52–72	In this paper we argue that it is possible and desirable to combine the easy input affordances of text with the powerful retrieval and visualization capabilities of graphical applications. We present WenSo, a tool that which uses lightweight text input to capture richly structured information for later retrieval and navigation in a graphical environment.
Rambling Enron E-mail	6 paragraphs, 24 sentences, 406 words	78%	\$9.72, 362 workers	44–52	A previous board member, Steve Burleigh, created our web site last year and gave me alot of ideas. For this year, I found a web site called eTeamZ that hosts web sites for sports groups. Check out our new page: [...]
Technical Writing ¹	3 paragraphs, 13 sentences, 291 words	82%	\$4.84, 188 workers	132–489	Figure 3 shows the pseudocode that implements this design for Lookup. FAWN-DS extracts two fields from the 160-bit key: the i low order bits of the key (the index bits) and the next 15 low order bits (the key fragment).

The Example Output column contains example edits from each input.

This wait time can be much longer because tasks can stall waiting for workers, as Table 1 shows.

Considering only work time and assuming negligible wait time, Shortn produced cuts within minutes. We again estimate overall work time by examining the median amount of time a worker spent in each stage of the Find-Fix-Verify process. This process reveals that the median shortening took 118 seconds of work time, or just under 2 minutes, when summed across all three stages ($Q_1 = 60$ seconds, $Q_3 = 3.6$ minutes). Using recruitment techniques developed since this research was published, users may see shortening tasks approaching a limit of 2 minutes.

The average paragraph cost \$1.41 to shorten under our pay model. This cost split into \$0.55 to identify an average of two patches, then \$0.48 to generate alternatives and \$0.38 to filter results for each of those patches. Our experience is that paying less slows down the later parts of the process, but it does not impact quality¹³—it would be viable for shortening paragraphs under a loose deadline.

Qualitatively, Shortn was most successful when the input had unnecessary text. For example, with the Blog input, Shortn was able to remove several words and phrases without changing the meaning of the sentence. Workers were able to blend these cuts into the sentence easily. Even the most technical input texts had extraneous phrases, so Shortn was usually able to make at least one small edit of this nature in each paragraph. As Soylent runs, it can collect a large database of these straightforward rewrites, then use them to train a machine learning algorithm to suggest some shortenings automatically.

Shortn occasionally introduced errors into the paragraph. While workers tended to stay away from cutting material they did not understand, they still occasionally flagged such patches. As a result, workers sometimes made edits that were grammatically appropriate but stylistically incorrect. For example, it may be inappropriate to remove the academic signaling phrase “In this paper we argue that...” from an introduction. Cuts were a second source of error: workers in the Fix stage would vote that a patch could be removed entirely from the sentence, but were not given the chance to massage the effect of the cut into the sentence. So, cuts often led to capitalization and punctuation problems at sentence boundaries. Modern auto-correction techniques could catch many of these errors. Parallelism was another source of error: for example, in Technical Writing (Table 1), the two cuts were from two different patches, and thus handled by separate workers. These workers could not predict that their cuts would not match, one cutting the parenthetical and the other cutting the main phrase.

To investigate the extent of these issues, we coded all 126 shortening suggestions as to whether they led to a grammatical error. Of these suggestions, 37 suggestions were ungrammatical, again supporting our rule of thumb that 30% of raw worker edits will be noisy. The Verify step caught 19 of the errors (50% of 37) while also removing 15 grammatical sentences. Its error rate was thus $(18 \text{ false negatives} + 15 \text{ false positives})/137 = 26.1\%$, again near 30%. Microsoft Word’s grammar checker caught 13 of the errors. Combining Word and Shortn caught 24 of the 37 errors.

We experimented with feeding the shortest output from the Blog text back into the algorithm to see if it could continue shortening. It continued to produce cuts between

70%–80% with each iteration. We ceased after 3 iterations, having shortened the text to less than 50% length without sacrificing much by way of readability or major content.

6.2. Crowproof evaluation

To evaluate Crowproof, we obtained a set of five input texts in need of proofreading. These inputs were error-ridden text that passes Word’s grammar checker, text written by an ESL student, quick notes from a presentation, a poorly written Wikipedia article (Dandu Monara), and a draft UIST paper. We manually labeled all spelling, grammatical and style errors in each of the five inputs, identifying a total of 49 errors. We then ran Crowproof on the inputs using a 20-minute stage timeout, with prices \$0.06 for Find, \$0.08 for Fix, and \$0.04 for Verify. We measured the errors that Crowproof caught, that Crowproof fixed, and that Word caught. We ruled that Crowproof had caught an error if one of the identified patches contained the error.

Results. Soylent’s proofreading algorithm caught 33 of the 49 errors (67%). For comparison, Microsoft Word’s grammar checker found 15 errors (30%). Combined, Word and Soylent flagged 40 errors (82%). Word and Soylent tended to identify different errors, rather than both focusing on the easy and obvious mistakes. This result lends more support to Crowproof’s approach: it can focus on errors that automatic proofreaders have not already identified.

Crowproof was effective at fixing errors that it found. Using the Verify stage to choose the best textual replacement, Soylent fixed 29 of the 33 errors it flagged (88%). To investigate the impact of the Verify stage, we labeled each unique correction that workers suggested as grammatical or not. Fully 28 of 62 suggestions, or 45%, were ungrammatical. The fact that such noisy suggestions produced correct replacements again suggests that workers are much better at verification than they are at authoring.

Crowproof’s most common problem was missing a minor error that was in the same patch as a more egregious error. The four errors that Crowproof failed to fix were all contained in patches with at least one other error; Lazy Turkers fixed only the most noticeable problem. A second problem was a lack of domain knowledge: in the ESL input, workers did not know what a GUI was, so they could not know that the author intended “GUIs” instead of “GUI.” There were also stylistic opinions that the original author might not have agreed with: in the Draft UIST input, the author had a penchant for triple dashes that the workers did not appreciate.

Crowproof shared many running time characteristics with Shortn. Its median work time was 2.8 minutes ($Q_1 = 1.7$ minutes, $Q_3 = 4.7$ minutes), so it completes in very little work time. Similarly to Shortn, its wait time was 18 minutes (Median = 17.6, $Q_1 = 9.8$, $Q_3 = 30.8$). It cost more money to run per paragraph ($\mu = \$3.40$, $\sigma = \$2.13$) because it identified far more patches per paragraph: we chose paragraphs in dire need of proofreading.

6.3. Human macro evaluation

We were interested in understanding whether end users could instruct Mechanical Turk workers to perform open-ended tasks. Can users communicate their intention clearly? Can workers execute the amateur-authored tasks correctly?

Method. We generated five feasible Human Macro scenarios. These scenarios included changing the tense of a story from past tense to present tense, finding a Creative Commons-licensed image to illustrate a paragraph, giving feedback on a draft blog post, gathering BibTeX for some citations, and filling out mailing addresses in a list. We recruited two sets of users: five undergraduate and graduate students in our computer science department (4 males) and five administrative associates in our department (all females). We showed each user one of the five prompts, consisting of an example input and output pair. We purposefully did not describe the task to the participants so that we would not influence how they wrote their task descriptions. We then introduced participants to The Human Macro and described what it would do. We asked them to write a task description for their prompt using The Human Macro. We then sent the description to Mechanical Turk and requested that five workers complete each request. In addition to the ten requests generated by our participants, one author generated five requests himself to simulate a user who is familiar with Mechanical Turk.

We coded results using two quality metrics: intention (did the worker understand the prompt and make a good faith effort?) and accuracy (was the result flawless?). If the worker completed the task but made a small error, the result was coded as good intention and poor accuracy.

Results. Users were generally successful at communicating their intention. The average command saw an 88% intention success rate (max = 100%, min = 60%). Typical intention errors occurred when the prompt contained two requirements: for example, the Figure task asked both for an image and proof that the image is Creative Commons-licensed. Workers read far enough to understand that they needed to find a picture, found one, and left. Successful users clearly signaled Creative Commons status in the title field of their request.

With accuracy, we again see that roughly 30% of work contained an error. (The average accuracy was 70.8%.) Workers commonly got the task mostly correct, but failed on some detail. For example, in the Tense task, some workers changed all but one of the verbs to present tense, and in the List Processing task, sometimes a field would not be correctly capitalized or an Eager Beaver would add too much extra information. These kinds of errors would be dangerous to expose to the user, because the user might likewise not realize that there is a small error in the work.

7. DISCUSSION

This section reviews some fundamental questions about the nature of paid, crowd-powered interfaces as embodied in SoyLent. Our work suggests that it may be possible to transition from an era where Wizard of Oz techniques were used only as prototyping tools to an era where a “Wizard of Turk” can be permanently wired into a system. We touch on resulting issues of wait time, cost, legal ownership, privacy, and domain knowledge.

In our vision of interface outsourcing, authors have immediate access to a pool of human expertise. Lag times in our current implementation are still on the order of minutes to hours, due to worker demographics, worker availability, the relative attractiveness of our tasks, and so on. While future growth in crowdsourced work will likely shorten lag

times, this is an important avenue of future work. It may be possible to explicitly engineer for responsiveness in return for higher monetary investment, or to keep workers around with other tasks until needed.²

With respect to cost, SoyLent requires that authors pay all workers for document editing—even if many changes never find their way into the final work product. One might therefore argue that interface outsourcing is too expensive to be practical. We counter that in fact all current document processing tasks also incur significant cost (in terms of computing infrastructure, time, software and salaries); the only difference is that interface outsourcing precisely quantifies the price of each small unit of work. While payment-per-edit may restrict deployment to commercial contexts, it remains an open question whether the gains in productivity for the author are justified by the expense.

Regarding privacy, SoyLent exposes the author’s document to third party workers without knowing the workers’ identities. Authors and their employers may not want such exposure if the document’s content is confidential or otherwise sensitive. One solution is to restrict the set of workers that can perform tasks: for example, large companies could maintain internal worker pools. Rather than a binary opposition, a continuum of privacy and exposure options exists.

SoyLent also raises questions over legal ownership of the resulting text, which is part-user and part-Turker generated. Do the Turkers who participate in Find-Fix-Verify gain any legal rights to the document? We believe not: the Mechanical Turk worker contract explicitly states that it is work-for-hire, so results belong to the requester. Likewise with historical precedent: traditional copyeditors do not own their edits to an article. However, crowd-sourced interfaces will need to consider legal questions carefully.

It is important that the research community ask how crowd-sourcing can be a social good, rather than a tool that reinforces inequality. Crowdsourcing is a sort of renewal of scientific management. Taylorism had positive impacts on optimizing workflows, but it was also associated with the dehumanizing elements of factory work and the industrial revolution. Similarly, naive crowdsourcing might treat people as a new kind of abstracted API call, ignoring the essential humanness of these sociotechnical systems. Instead, we need to evolve our design process for crowdsourcing systems to involve the crowds workers’ perspective directly.

8. CONCLUSION

The following conclusion was Shortn’ed to 85% length: This chapter presents SoyLent, a word processing interface that uses crowd workers to help with proofreading, document shortening, editing and commenting tasks. SoyLent is an example of a new kind of interactive user interface in which the end user has direct access to a crowd of workers for assistance with tasks that require human attention and common sense. Implementing these kinds of interfaces requires new software programming patterns for interface software, since crowds behave differently than computer systems. We have introduced one important pattern, Find-Fix-Verify, which splits complex editing tasks into a series of identification, generation, and verification stages that use independent agreement and voting produce reliable results. We evaluated SoyLent with a range of editing tasks, finding and correcting 82% of grammar

errors when combined with automatic checking, shortening text to approximately 85% of original length per iteration, and executing a variety of human macros successfully.

Future work falls in three categories. First are new crowd-driven features for word processing, such as readability analysis, smart find-and-replace (so that renaming “Michael” to “Michelle” also changes “he” to “she”) and figure or citation number checking. Second are new techniques for optimizing crowd-programmed algorithms to reduce wait time and cost. Finally, we believe that our research points the way toward integrating on-demand crowd work into other authoring interfaces, particularly in creative domains like image editing and programming. 

References

- Andersen, D.G., Franklin, J., Kaminsky, M., Phanishayee, A., Tan, L., Vasudevan, V. FAWN: A fast array of wimpy nodes. In *Proc. SOSP '09* (2009).
- Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatrowicz, A., White, B., White, S., Yeh, T. VizWiz: Nearly real-time answers to visual questions. In *Proc. UIST '10* (2010).
- Clarke, J., Lapata, M. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proc. ACL '06* (2006).
- Cypher, A. *Watch What I Do*. MIT Press, Cambridge, MA, 1993.
- Dourish, P., Bellotti, V. Awareness and coordination in shared workspaces. In *Proc. CSCW '92* (1992).
- Heer, J., Bostock, M. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In *Proc. CHI '10* (2010).
- Ishii, H., Ullmer, B. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proc. UIST '97* (1997).
- Kittur, A., Chi, E.H., Suh, B. Crowdsourcing user studies with Mechanical Turk. In *Proc. CHI '08* (2008).
- Kukich, K. Techniques for automatically correcting words in text. *ACM Comput. Surv. (CSUR)* 24, 4 (1992), 377–439.
- Le, J., Edmonds, A., Hester, V., Biewald, L. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Proc. SIGIR 2010 Workshop on Crowdsourcing for Search Evaluation* (2010), 21–26.
- Little, G., Chilton, L., Goldman, M., Miller, R.C. TurkKit: Human computation algorithms on Mechanical Turk. In *Proc. UIST '10* (2010) ACM Press.
- Marcu, D. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA, 2000.
- Mason, W., Watts, D.J. Financial incentives and the performance of crowds. In *Proc. HCOMP '09* (2009) ACM Press.
- Quinn, A.J., Bederson, B.B. Human computation: A survey and taxonomy of a growing field. In *Proc. CHI '11* (2011) ACM.
- Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., Tomlinson, B. Who are the crowdworkers? Shifting demographics in Amazon Mechanical Turk. In *alt.chi '10* (2010) ACM Press.
- Sala, M., Partridge, K., Jacobson, L., Begole, J. An exploration into activity-informed physical advertising using PEST. In *Pervasive '07*, volume 4480 of *Lecture Notes in Computer Science* (Berlin, Heidelberg, 2007), Springer, Berlin, Heidelberg.
- Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y. Cheap and fast—But is it good? Evaluating non-expert annotations for natural language tasks. In *Proc. ACL '08* (2008).
- Sorokin, A., Forsyth, D. Utility data annotation with Amazon Mechanical Turk. *Proc. CVPR '08* (2008).
- von Ahn, L., Dabbish, L. Labeling images with a computer game. In *CHI '04* (2004).

Michael S. Bernstein (msb@cs.stanford.edu) Stanford University, Stanford, CA.

Robert C. Miller (rcm@mit.edu) MIT CSAIL, Cambridge, MA.

Björn Hartmann (bjoern@cs.berkeley.edu) Computer Science Division University of California, Berkeley, CA.

David R. Karger (karger@mit.edu) MIT CSAIL, Cambridge, MA.

Greg Little (glittle@csail.mit.edu)

David Crowell (dcrowell.mit@gmail.com)

Mark S. Ackerman (ackerm@umich.edu) Computer Science & Engineering University of Michigan, Ann Arbor, MI.

Katrina Panovich (kpan@google.com), Google, Inc., Mountain View, CA.

© 2015 ACM 0001-0782/15/08 \$15.00



Watch the authors discuss their work in this exclusive *Communications* video.
<http://cacm.acm.org/videos/soylent-a-word-processor-with-a-crowd-inside>

World-Renowned Journals from ACM

ACM publishes over 50 magazines and journals that cover an array of established as well as emerging areas of the computing field. IT professionals worldwide depend on ACM's publications to keep them abreast of the latest technological developments and industry news in a timely, comprehensive manner of the highest quality and integrity. For a complete listing of ACM's leading magazines & journals, including our renowned Transaction Series, please visit the ACM publications homepage: www.acm.org/pubs.

ACM Transactions on Interactive Intelligent Systems



ACM Transactions on Interactive Intelligent Systems (TIIS). This quarterly journal publishes papers on research encompassing the design, realization, or evaluation of interactive systems incorporating some form of machine intelligence.

ACM Transactions on Computation Theory



ACM Transactions on Computation Theory (ToCT). This quarterly peer-reviewed journal has an emphasis on computational complexity, foundations of cryptography and other computation-based topics in theoretical computer science.

PLEASE CONTACT ACM MEMBER SERVICES TO PLACE AN ORDER

Phone: 1.800.342.6626 (U.S. and Canada)
 +1.212.626.0500 (Global)

Fax: +1.212.944.1318
 (Hours: 8:30am–4:30pm, Eastern Time)

Email: acmhelp@acm.org

Mail: ACM Member Services
 General Post Office
 PO Box 30777
 New York, NY 10087-0777 USA



Association for
Computing Machinery

Advancing Computing as a Science & Profession

www.acm.org/pubs



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 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: August 31, 2015 (or until positions are filled).



a place of mind

THE UNIVERSITY OF BRITISH COLUMBIA

ENDOWED CHAIR IN COMPUTER SYSTEMS

The Department of Computer Science at the University of British Columbia (UBC) Vancouver Campus, is seeking candidates with exceptional scientific records in the area of Computer Systems, broadly defined, for a fully endowed chair at the rank of tenured Professor, made possible by a generous donation to the department by David Cheriton, a distinguished Stanford Professor and UBC alumnus. UBC Computer Science (www.cs.ubc.ca) ranks among the top departments in North America, with 54 tenure-track faculty, 200 graduate students, and 1500 undergraduates.

The successful candidate must demonstrate sustained research and teaching excellence judged by the following key factors: i) publication record in the highest caliber international computer science systems, networking, operating systems, distributed systems, and security conferences and journals; ii) impact on the field and/or industry resulting from his or her publications, or resulting from other research artifacts such as software; iii) successful mentorship of graduate students, and collaboration with other researchers in the field; iv) teaching of both undergraduate and graduate courses and department service; and v) external funding and leadership within his or her research community. Preference will be given to candidates that have built or contributed to the building of real software systems of note. Outstanding industrial researchers are also encouraged to apply—all evidence of a candidate's public speaking, teaching and mentoring effectiveness, such as in seminars, tutorials, or student project supervision, will be considered. The potential of an applicant's research program to complement and extend existing research strengths of the department and the University will be an important factor in selection. The anticipated start date is July 1, 2016.

Applicants must submit a CV, a research statement, a teaching statement, and the names of at least four references. The teaching statement should include a record of teaching interests and experience. Applications may be submitted online at <https://apps.cs.ubc.ca/fac-recruit/systems/apply/form.jsp>.

The website will remain open for submissions through the end of the day on September 1st, 2015. The website may remain open past that date at the discretion of the recruiting committee. All applications submitted while the website remains open will be considered.

UBC hires on the basis of merit and is strongly committed to equity and diversity within its community. We especially welcome applications from members of visible minority groups, women, Aboriginal persons, persons with disabilities, persons of minority sexual orientations and gender identities, and others with the skills and knowledge to engage productively with diverse communities. All qualified candidates are encouraged to apply; however, Canadian citizens and permanent residents will be given priority.

If you have questions about the application process, please contact the Chair of the Systems Recruiting Subcommittee by email at fac-rec-systems@cs.ubc.ca

Norm Hutchinson
Chair, Systems Recruiting Subcommittee
Department of Computer Science
University of British Columbia
Vancouver BC V6T 1Z4
Canada

Please do not email applications. Apply online via: <https://apps.cs.ubc.ca/fac-recruit/systems/apply/form.jsp>

Boise State University

Department of Computer Science

Eight Open Rank, Tenured/Tenure-Track Faculty Positions

The Department of Computer Science at Boise State University invites applications for eight open rank, tenured/tenure-track faculty positions. Seeking applicants in the areas of big data (including distributed systems, HPC, machine learning, visualization), cybersecurity, human computer interaction and computer science education research. Strong applicants from other areas of computer science will also be considered.

Applicants should have a commitment to excellence in teaching, a desire to make significant contributions in research, and experience in collaborating with faculty and local industry to develop and sustain funded research programs. A PhD in Computer Science or a closely related field is required by the date of hire. For additional information, please visit <http://coen.boisestate.edu/cs/jobs>.

California State University, East Bay (Hayward, CA)

Department of Computer Science

Faculty Position in Computer Science

2 POSITIONS (OAA Position No 15-16 CS-DATA/CLOUD/CORE-TT) The Department invites applications for 2 tenure-track appointments as Assistant Professor in Computer Science (considering all areas of computer science, capable of teaching in emerging areas) starting Fall 2016. For details see <http://www20.csueastbay.edu/about/career-opportunities/>. For questions, email: cssearch@mcs.csueastbay.edu.



ADVERTISING IN CAREER OPPORTUNITIES

How to Submit a Classified Line Ad: Send an e-mail to acmmidiasales@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 be 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: acmmidiasales@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
acmmidiasales@acm.org



DOI:10.1145/2791401

Dennis Shasha

Upstart Puzzles Brighten Up

YOU ARE GIVEN two bags, each containing some number $NumPerBag$ of flares. You know there are $NumBad$ flares in one of the bags but not which bag. The other bag has all good flares. Each time you test a flare, you use it up.

Here is the first challenge: You want to take $NumBad-1$ flares with you and want to know all are good. Further, you want to use up as few flares as possible in the process. It is fine that when you are done, you may not know which of the unused flares you leave behind are good.

Warm-up. If all the flares in the bad bag are indeed bad, then how many flares would you need to test?

Solution to this warm-up. Test just one flare in one bag. After that you would have at least $NumPerBag-1 = NumBad-1$ good flares to take from the bag you know has only good flares.

For general values of $NumPerBag$ and $NumBad$, consider two strategies:

Balanced. Take a flare from the first bag and test it, then one from the other bag and test it, and continue alternating until you find a bad one or you reach $NumBad-1$ in one of the bags, in which case you know the remaining ones in that bag are good; and

Unbalanced. Keep taking one flare from a single bag and testing it until you find a bad one or reach $NumBad-1$ in that bag.

Which strategy uses up fewer flares in the worst case?

The average case is worthy of an upstart challenge:

Upstart 1. Assuming $NumBad = 3$, are there values of $NumPerBag$ for which the balanced strategy uses up fewer flares than the unbalanced strategy on average, assuming no particular ordering of the flares in either bag? And vice versa. Is there some mixed strategy that is better than either one alone?

2. Now imagine you want to take $NumBad$ flares (not just $NumBad-1$) on your trip with the guarantee all are good. Which strategy would give you the best chance of achieving this? For this challenge, the number of flares you use up in testing is unimportant.

Upstart 2. Given $NumPerBag$ and $NumBad$, suppose you want to take d more than $NumBad$ with the guarantee all are good. What is the best strategy to use (it may be a hybrid), and what probability of success as a function of d can you achieve?

Solution to 1, or taking $NumBad - 1$ good flares on the trip. Unbalanced, because it will use up $1 + NumPerBag - NumBad$ flares in the worst case, whereas balanced may use up $1 + 2*(NumPerBag - NumBad)$ in the worst case.

Solution to 2, or taking $NumBad$ good flares on the trip. With the unbalanced strategy, if you are lucky enough to start testing flares with the bad bag, you will be able to take $NumBad$ good flares for sure. If you start testing flares with the good bag, then you will test $1 + NumPerBag - NumBad$ flares from that good bag, leaving you $NumBad - 1$ good flares. But you can get one more good flare from the bad bag by testing flares from that bag until you discover all $NumBad$ flares and seeing whether you have any more flares left in that bag. With the balanced strategy, if you discover no bad flare by the time both bags have $NumBad$ flares left and the next flare tested is also good, then the strategy fails to deliver $NumBad$ flares that are all good. However, that is the only case in which the balanced strategy loses. Balanced is thus better. ■



Contemplating two bags containing flares, only one of which with bad flares; the goal is to take only good flares, which could come from one or from both bags, on a trip.

All are invited to submit their solutions to upstartpuzzles@cacm.acm.org; solutions to upstarts and discussion will be posted at <http://cs.nyu.edu/cs/faculty/shasha/papers/cacmpuzzles.html>

Dennis Shasha (dennishasha@yahoo.com) is a professor of computer science in the Computer Science Department of the Courant Institute at New York University, New York, as well as the chronicler of his good friend the omniheurist Dr. Ecco.

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

Computing Reviews

Connect with our Community of Reviewers

“Computing Reviews gives much needed broad perspective on the entire computing field (and its periphery). It also makes one aware of the State of the Art.”

- George Hacken



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

